Unclassified
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION Naval Postgraduate School 6b. OFFICE SYMBOL (If applicable) 32	7a. NAME OF MONITORING ORGANIZATION Naval Postgraduate School
6c. ADDRESS (City, State, and ZIP Code) Monterey, CA 93943-5000	7b. ADDRESS (City, State, and ZIP Code) Monterey, CA 93943-5000
8a. NAME OF FUNDING/SPONSORING ORGANIZATION 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State, and ZIP Code)	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT ACCESSION NO.

11. TITLE (Include Security Classification)
The Design of a Robust Autopilot for the Archytas Prototype via Linear Quadratic Synthesis

12. PERSONAL AUTHOR(S)
Joseph P. Davis

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year,Month,Day)	15. PAGE COUNT
Master's Thesis	FROM _____ TO _____	December 1992	132

16. SUPPLEMENTARY NOTATION
The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	
			Linear Quadratic Regulator, AROD, Archytas, Reduced Order, performance measure

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

The purpose of this research is to design, simulate and implement a robust autopilot for the vertical mode of operation of the Archytas prototype. Archytas is an Unmanned Air Vehicle that is designed to take off and land vertically, and to transition to horizontal forward flight. A feedback control scheme is designed for both the single-input, single-output and the multi-input, multi-output subsytems using optimal control techniques. In this research, the linear quadratic regulator performance measure is modified to allow for its application to the tracking problem solution. Additionally, the control systems are designed using reduced order models. Computer simulations show that the reduced order controller designs provide results comparable to the full order controller designs. Successful hardware tests with the roll rate control system validated the reduced order model design philosophy used in this research.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT [X] UNCLASSIFIED/UNLIMITED [ ] SAME AS RPT. [ ] DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION Unclassified
22a. NAME OF RESPONSIBLE INDIVIDUAL Jeffrey B. Burl	22b. TELEPHONE (Include Area Code) (408) 646 - 2390 22c. OFFICE SYMBOL EC/Bl

DD Form 1473, JUN 86 Previous editions are obsolete. SECURITY CLASSIFICATION OF THIS PAGE
S/N 0102-LF-014-6603 Unclassified

The Design of a Robust Autopilot for the Archytas Prototype via Linear Quadratic Synthesis

by

Joseph P. Davis
Captain, United States Marine Corps
B.E., University of Mississippi, 1986

Submitted in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
December 1992

ABSTRACT

The purpose of this research is to design, simulate and implement a robust autopilot system for the vertical mode of operation of the Archytas prototype. Archytas is an Unmanned Air Vehicle that is designed to take off and land vertically, and to transition to horizontal forward flight. A feedback control scheme is designed for both the single-input, single-output and the multi-input, multi-output subsystems using optimal control techniques. In this research, the linear quadratic regulator performance measure is modified to allow for its application to the tracking problem solution. Additionally, the control systems are designed using reduced order models. Computer simulations show that the reduced order controller designs provide results comparable to the full order controller designs. Successful hardware tests with roll rate control system validated the reduced order model design philosophy used in this research.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGMENTS

I would like to thank Professor Jeff Burl for sharing his insight, patience and instructive guidance as my thesis advisor. I would also like to thank Professor Rick Howard for allowing me the opportunity to work on the Archytas project. It is with deepest gratitude that I thank my loving wife, Andrea, for her unfailing support and understanding over the past two years. I dedicate this thesis to you.

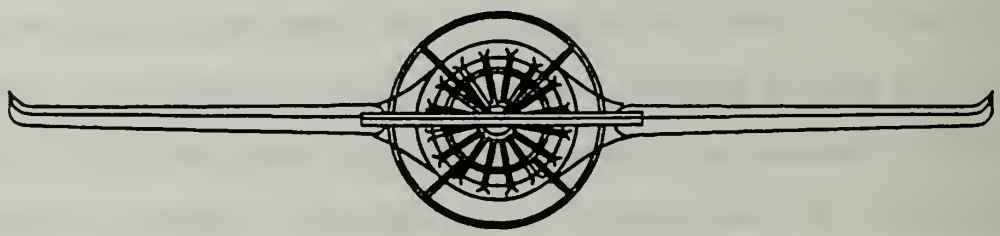
I. INTRODUCTION

A. THE ARCHYTAS CONCEPT

The current inventory of Unmanned Air Vehicles (UAVs) is not able to meet the expanding need for real-time intelligence at the Marine expeditionary unit or Army battalion level. The Naval Postgraduate School UAV Flight Research Lab is directing efforts at developing a ducted-fan vertical takeoff and landing (VTOL) vehicle to meet these increasing needs. The NPS air vehicle, named Archytas, is serving as a technology demonstrator to evaluate the concept of a winged ducted-fan VTOL aircraft. The research is being directed at applying the technology and equipment developed in the U.S. Marine Corps' Airborne Remotely Operated Device (AROD) program and the U.S. Army's AQUILA program.

Archytas, pictured in Figure 1.1, is designed to take off and land vertically. After climbing to altitude, Archytas will transition to horizontal flight by pitching about its center of gravity to a wings level attitude. The positioning of the duct and wings (including the canard) allow for the vertical takeoff and landing capability. The ability to transition to horizontal flight will extend the vehicle's range and provide the capability for a high speed dash.

TOP VIEW



SIDE VIEW

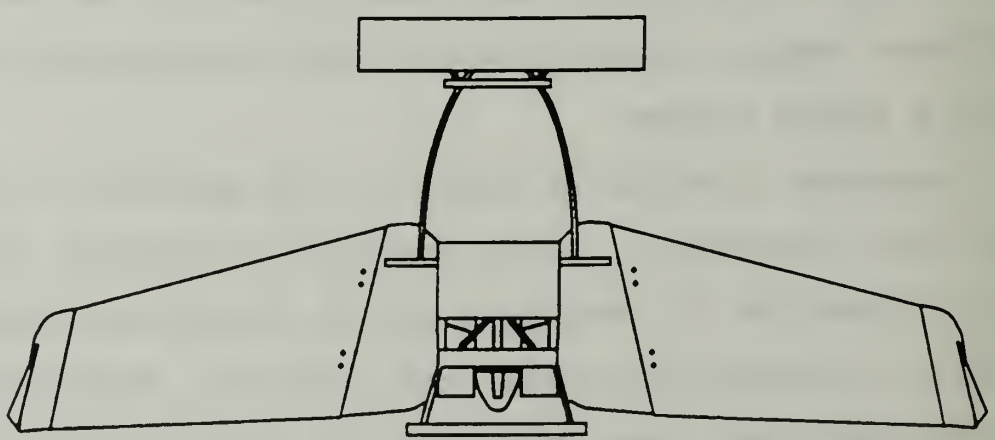


Figure 1.1 Sketch of Archytas

B. THE CONTROL PROBLEM

Archytas is powered by a vertically mounted 28-horsepower engine turning a three-bladed propeller. The use of a single propeller in a duct (ducted fan) simplifies the design, but intensifies the stability and control problems. The dynamic behavior about a given axis is coupled with other vehicle dynamics. In particular, there are three types of coupling:

1. The single propeller design introduces a gyroscopic coupling between the pitch and yaw axes.
2. Reactive torques are applied to the roll axis as the engine speed is varied.
3. A loss of lift due to thrust occurs when the vehicle is pitched during the translation to horizontal flight.

It was the goal of this thesis to design a control system which would allow stable flight of Archytas during takeoff and landing, and during the vertical mode of operation. Because of the coupled nature of the Archytas control problem, linear quadratic regulator control theory was used.

C. THESIS ORGANIZATION

In Chapter II, the general nonlinear equations of motion are developed. These equations of motion are then applied to Archytas with special attention given to the effects of the spinning propeller. Additionally, the equations of motion are linearized about the hover operating condition using the small-disturbance theory.

Chapter III discusses the state space representation of a general system and develops a procedure for selecting state variables for the optimal control tracking problem. Key to the design of the Archytas control system is the proper selection of the state variables.

In Chapter IV, three control law designs are formulated based on the linearized hover equations. One design is for the single input, single output (SISO) roll rate controller. The second design is for the SISO altitude rate controller. These two SISO systems are similar in their development. Next, the multiple input, multiple output (MIMO) pitch and yaw angle controller is designed. Central to each control law design is the use of a reduced order model to simplify the design process and physical implementation. Finally, these control laws are applied to the nonlinear model for validation.

Chapter V discusses the field test results of the roll rate controller. The roll rate controller was evaluated with Archytas mounted on a test stand to allow a roll about the longitudinal axis. In addition, conclusions based on the computer simulations, the field tests, and recommendations for future research are presented.

II. MODELING THE ARCHYTAS PROTOTYPE

The purpose of this chapter is to develop a suitable dynamic model of the Archytas prototype. Because Archytas is a ducted fan device, special attention must be given to the significant gyroscopic contribution of its propeller.

A. DERIVATION OF RIGID BODY EQUATIONS OF MOTION

The rigid body equations of motion in this section are developed for the Archytas prototype in the following way. First, Archytas is regarded as a single rigid body, and the equations of motion are derived with respect to a set of body fixed axes. These equations are the general equations governing aerodynamic flight for all aircraft. Next, the changes introduced by the spinning rotor are evaluated and included in the equations. These are the gyroscopic effects due to the propeller. Finally, the development of a complete model is undertaken for the specific case of Archytas using the actual measurements and experimental data from the AROD prototype as first approximations.

1. Force and Moment Equations

The general equations of motion are developed for a typical aircraft in References 1 and 2. A combination of the two approaches is taken here to arrive at the set of equations describing Archytas. The equations of motion are

obtained from Newton's second law, which states: The summation of all external forces acting on a body is equal to the time rate of change of the momentum of the body; and the summation of the external moments acting on the body is equal to the time rate of change of the moment of the momentum (angular momentum). The time rates of change of linear and angular momentum are referred to an absolute or inertial reference frame. This absolute or inertial reference frame is an axis system fixed to the Earth. Figure 2.1 depicts both the body fixed axes and the inertial reference frame.[Ref. 1: p.84]

Newton's second law can be expressed in the following vector equations:

$$\sum \underline{F} = \frac{d}{dt}(m\underline{v}) \; ; \tag{2.1}$$

$$\sum \underline{M} = \frac{d}{dt}\underline{H} \; ; \tag{2.2}$$

where $\underline{F}$ is the externally applied force, $\underline{M}$ the externally applied moment about the center of mass, $\underline{v}$ the velocity vector, and $\underline{H}$ the angular momentum vector about the center of mass.

The vector equations, in scalar form, consist of three force equations and three moment equations. The force equations can be expressed as

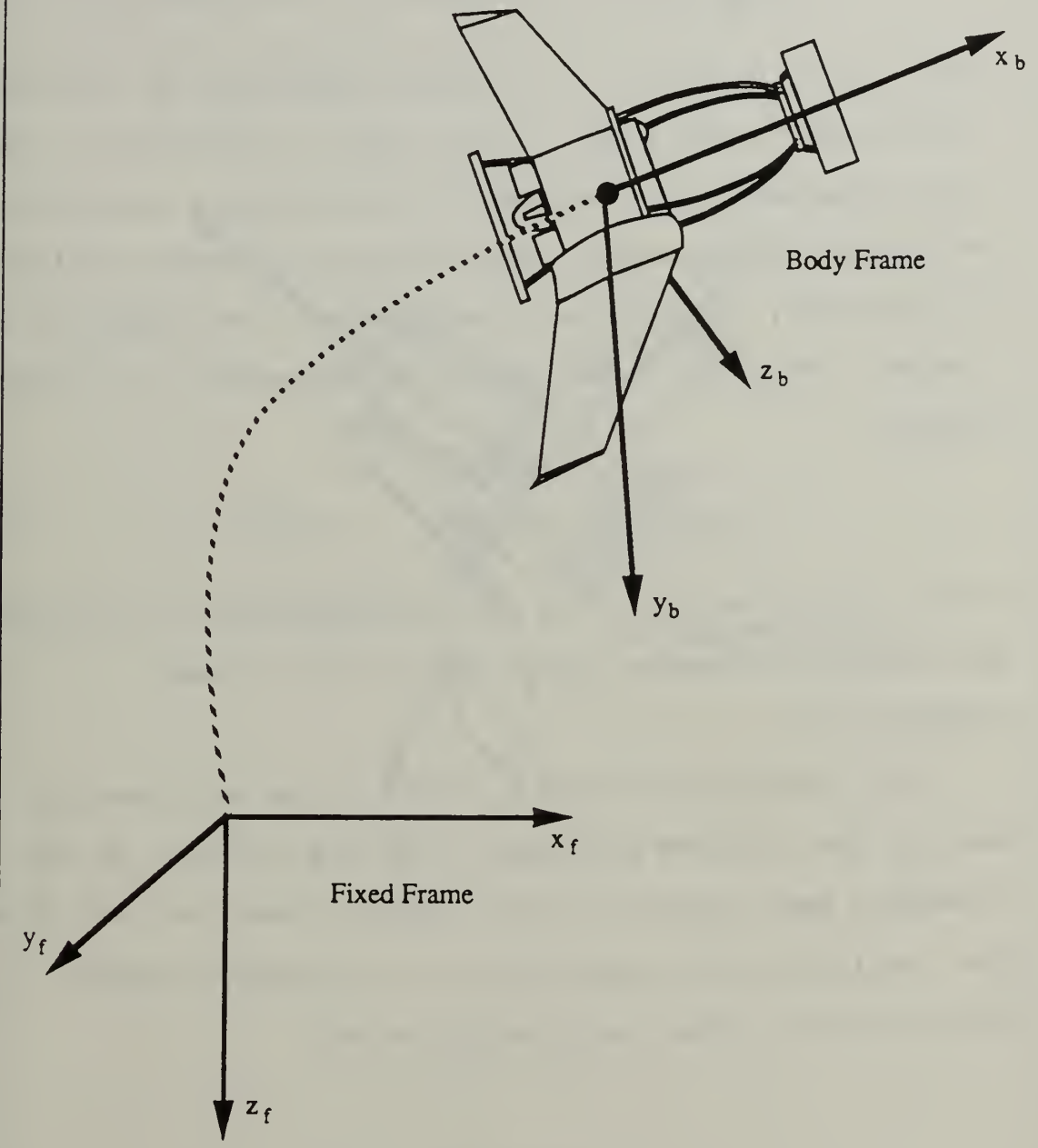


Figure 1

Figure 2.1 Body and Fixed Axes System

$$F_x=\frac{d}{dt}(mu)\ ; \qquad F_y=\frac{d}{dt}(mv)\ ; \qquad F_z=\frac{d}{dt}(mw)\ ; \tag{2.3}$$

where F_x, F_y, F_z and u, v, w are the components of the force and velocity along the x, y and z axes, respectively. The force components are composed of contributions due to the aerodynamic, propulsive, and gravitational forces acting on the aircraft, the Archytas prototype for the purpose of this thesis. The moment equations can be expressed in a similar manner:

$$L=\frac{d}{dt}H_x\ ; \qquad M=\frac{d}{dt}H_y\ ; \qquad N=\frac{d}{dt}H_z\ ; \tag{2.4}$$

where L, M, N and H_x, H_y, H_z are the components of the moment and moment of momentum along the x, y and z axes respectively.

Now, considering Figure 2.2, let δm be an element of mass of the Archytas prototype, $\underline{v}$ be the velocity of the elemental mass relative to the inertial axes, and let $\delta\underline{F}$ be the resultant force that acts upon it. Newton's second law then gives the equation of motion of δm:

$$\delta\underline{F}=\delta m\frac{d\underline{v}}{dt}\ . \tag{2.5}$$

The total force acting on the vehicle is a summation of all the forces that act upon all the elements. The internal

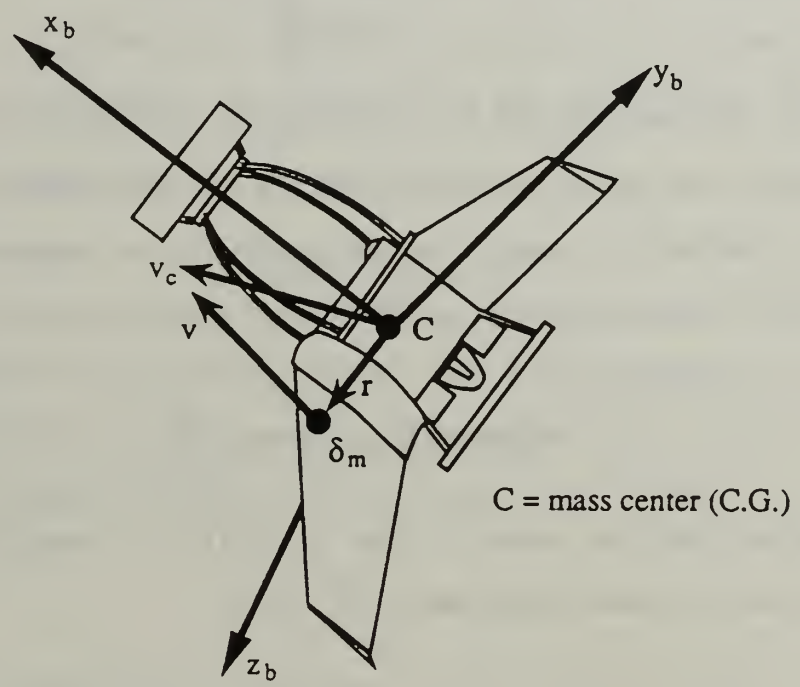


Figure 2.2 An Element of Mass on the Archytas

forces, those exerted by one element upon another, all occur in equal and opposite pairs, by Newton's third law, and hence contribute nothing to the summation. Thus $\Sigma\delta\underline{F}=\underline{F}$ is the resultant external force acting upon the vehicle. The velocity of the differential mass δm is given by

$$\underline{v}=\underline{v}_c+\frac{d\underline{r}}{dt} \; ; \tag{2.6}$$

were $\underline{v}_c$ is the velocity of the center of mass of the aircraft and $d\underline{r}/dt$ is the velocity of the element relative to the center of mass. Substituting this expression for the velocity into Newton's second law, equation (2.1), yields

$$\Sigma\delta\underline{F}=\underline{F}=\sum \delta m\frac{d}{dt}\left(\underline{v}_c+\frac{d\underline{r}}{dt}\right) \; . \tag{2.7}$$

Assuming that the mass of the aircraft is constant, equation (2.7) can be rewritten as

$$\underline{F}=\frac{d}{dt}\sum \left(\underline{v}_c+\frac{d\underline{r}}{dt}\right)\delta m \tag{2.8}$$

or

$$\underline{F}=m\frac{d\underline{v}_c}{dt}+\frac{d^2}{dt^2}\sum \underline{r}\delta m \; . \tag{2.9}$$

Because $\underline{r}$ is measured from the center of mass, the summation $\Sigma\underline{r}\delta m$ is equal to zero and the force equation (2.9) becomes

$$\underline{F}=m\frac{d\underline{V}_c}{dt} \; ; \qquad (2.10)$$

which relates the external force on the aircraft to the motion of the vehicle's center of mass.

The relationship between the external moment and the rotation of the aircraft is obtained from a consideration of the moment of momentum. For the differential element of mass, δm, the moment of momentum is by definition $\delta\underline{H}=\underline{r}\times\underline{V}\delta m$. The moment equation can be written as

$$\delta\underline{M}=\frac{d}{dt}\delta\underline{H}=\frac{d}{dt}(\underline{r}\times\underline{V})\,\delta m \; . \qquad (2.11)$$

The velocity of the mass element can be expressed in terms of the velocity of the center of mass and the velocity of the mass element relative to the center of mass:

$$\underline{V}=\underline{V}_c+\frac{d\underline{r}}{dt}=\underline{V}_c+\underline{\omega}\times\underline{r} \; ; \qquad (2.12)$$

where $\underline{\omega}$ is the angular velocity of the vehicle and $\underline{r}$ is the position of the mass element measured from the center of mass. The total moment of momentum can be written as

$$\underline{H}=\sum\delta\underline{H}=\sum(\underline{r}\times\underline{V}_c)\,\delta m+\sum[\underline{r}\times(\underline{\omega}\times\underline{r})]\,\delta m \; . \qquad (2.13)$$

The velocity $\underline{V}_c$ is a constant with respect to the summation and can be taken outside of the summation sign

$$\underline{H}=(\sum \underline{r}\delta m) \times \underline{v}_c+\sum (\underline{r}\times\underline{\omega}\times\underline{r})\,\delta m \ . \qquad (2.14)$$

The first term in equation (2.14) is zero because the term $\sum\underline{r}\delta m=0$ as explained previously. The position vectors and angular velocity can be expressed as

$$\underline{r}=x\underline{i}+y\underline{j}+z\underline{k} \ ; \qquad (2.15)$$

$$\underline{\omega}=p\underline{i}+q\underline{j}+r\underline{k} \ ; \qquad (2.16)$$

where p, q and r are the scalar components of $\underline{\omega}$, and $\underline{i}$, $\underline{j}$, $\underline{k}$ are unit vectors in the directions of x, y and z. Substituting $\underline{\omega}$ and $\underline{r}$ into equation (2.14) and expanding, $\underline{H}$ can be written as

$$\underline{H}=(p\underline{i}+q\underline{i}+r\underline{k})\sum (x^2+y^2+z^2)\,\delta m-\sum (x\underline{i}+y\underline{i}+z\underline{k})\,(px+qy+rz)\,\delta m \ . \qquad (2.17)$$

The scalar components of $\underline{H}$ are

$$\begin{aligned} H_x &= p\sum(y^2+z^2)\ \delta m-q\sum xy\ \delta m-r\sum xz\ \delta m \ ; \\ H_y &= -p\sum xy\ \delta m+q\sum(x^2+z^2)\ \delta m\ -r\sum yz\ \delta m \ ; \\ H_z &= -p\sum xz\ \delta m-q\sum yz\ \delta m+r\sum(x^2+y^2)\ \delta m \ . \end{aligned} \qquad (2.18)$$

The summations in the above equations are the mass moment and products of inertia of the aircraft and are defined as

$$I_x=\iiint_V (y^2+z^2)\,\delta m\ , \qquad I_{xy}=\iiint_V xy\ \delta m\ ;$$

$$I_y=\iiint_V (x^2+z^2)\,\delta m\ , \qquad I_{xz}=\iiint_V xz\ \delta m\ ; \qquad (2.19)$$

$$I_z=\iiint_V (x^2+y^2)\,\delta m\ , \qquad I_{yz}=\iiint_V yz\ \delta m\ .$$

The terms I_x, I_y and I_z are the mass moments of inertia of the body about the x, y and z axes, respectively. The terms with the mixed indices are called the products of inertia. Both the moments and products of inertia depend on the shape of the body and the manner in which its mass is distributed. The larger the moments of inertia the greater the resistance the body will have to rotation. Applying the notation of equation (2.19) to equation (2.18), yields the scalar equations for the moment of momentum:

$$\begin{aligned} H_x &= pI_x-qI_{xy}-rI_{xz}\ ; \\ H_y &= -pI_{xy}+qI_y-rI_{yz}\ ; \qquad (2.20) \\ H_z &= -pI_{xz}-qI_{yz}+rI_z\ . \end{aligned}$$

If the reference frame is not fixed to the aircraft, then, as the aircraft rotates, the moments and products of inertia will vary with time. To avoid this difficulty an axis system will be fixed to the aircraft (body axis system). Now the derivatives of the vectors $\underline{v}$ and $\underline{H}$ referred to the rotating body frame of reference must be determined.

It can be shown that the derivatives of an arbitrary vector $\underline{A}$ referred to a rotating body frame having an angular velocity $\underline{\omega}$ can be represented by the following vector identity

$$\left.\frac{d\underline{A}}{dt}\right|_I = \left.\frac{d\underline{A}}{dt}\right|_B + \underline{\omega}\times\underline{A} \ ; \qquad \textbf{(2.21)}$$

where the subscripts I and B refer to the inertial and body fixed frames of reference, respectively. Applying this identity to equations (2.1) and (2.2) yields

$$\underline{F} = m\left.\frac{d\underline{V}_c}{dt}\right|_B + m(\omega\times\underline{V}_c) \ ; \qquad \textbf{(2.22)}$$

$$\underline{M} = \left.\frac{d\underline{H}}{dt}\right|_B + \omega\times\underline{H} \ . \qquad \textbf{(2.23)}$$

These are the general equations governing aerodynamic flight and have the scalar components:

$$F_x = m(\dot{u}+qw-rv), \quad F_y = m(\dot{v}+ru-pw), \quad F_z = m(\dot{w}+pv-qu); \qquad \textbf{(2.24)}$$

$$L = \dot{H}_x + qH_z - rH_y, \quad M = \dot{H}_y + rH_x - pH_z, \quad N = \dot{H}_z + pH_y - qH_x \ . \qquad \textbf{(2.25)}$$

The components of the force and moment acting on the aircraft are composed of aerodynamic, gravitational and propulsive contributions.

At this point, it is recognized that most aircraft have a plane of symmetry. If the xz plane is selected to coincide with this plane of symmetry, then from equation (2.19), $I_{xy}=I_{yz}=0$ must be satisfied. However, for the case of

the Archytas, the three blades of the propeller provide two planes of symmetry, xz and xy. Thus, the products of inertia, I_{xy}, I_{yz} and I_{xz}, equal zero. The moment equations for Archytas can now be written as

$$\begin{aligned} L &= I_x\dot{p} + qr(I_z - I_y) \ ; \\ M &= I_y\dot{q} + rp(I_x - I_z) \ ; \\ N &= I_z\dot{r} + pq(I_y - I_x) \ . \end{aligned} \qquad (2.26)$$

2. Effect of the Spinning Rotor

Archytas, like AROD, is a gyroscope. The single propeller rotates about the longitudinal vehicle axis to produce a downwash or jet of air through the duct which makes up the Archytas body. This spinning rotor exerts a gyroscopic moment on the body of the vehicle. Reference 3 states that in developing the equations of motion for aircraft with propellers which exert gyroscopic moments on the body "more often than not, such gyroscopic moments turn out to be negligible." However, as demonstrated by Bassett [Ref. 4: p. 19], in the AROD case the angular momentum of the propeller, $I_{p(AROD)}$, equals 11.3 ft^2-lb_m/sec. Compared with AROD's nominal total mass of 2.64 lb_m (85 lb_w), it is clear that the angular momentum imparted by the propeller is significant and that gyroscopic effects will play a large part in modeling the dynamic behavior of AROD. For Archytas, I_p is identically equal to $I_{p(AROD)}$, 11.3 ft^2-lb_m/sec. Compared with Archytas' nominal total mass of 3.11 lb_m (100

lb_w), it is clear that similar to AROD the gyroscopic effects will be significant and must be included in modeling the dynamic behavior of Archytas. This gyroscopic moment can be accounted for as follows.

Angular momentum, $\underline{H}_p$, due to the propeller is defined as

$$\underline{H}_p = I_p \omega_p = \underline{i} H_{px} + \underline{j} H_{py} + \underline{k} H_{pz} \ ; \qquad \textbf{(2.27)}$$

where I_p is the propeller moment of inertia, and ω_p is the propeller angular velocity. Since the propeller lies in the yz plane and spins symmetrically about the x axis, $\underline{H}_p$ is directed only along x and $H_{py}=H_{pz}=0$. Equation (2.27) becomes

$$\underline{H}_p = I_p \omega_p = i H_{px} \ . \qquad \textbf{(2.28)}$$

Etkin [Ref. 2: p.93] states that the resultant angular momentum of an aircraft with spinning propellers is obtained simply by adding $\underline{H}_p$ to the $\underline{H}$ previously defined by equation (2.20). Adding equations (2.28) and (2.20) and keeping in mind that the products of inertia equal zero, yields

$$\begin{aligned} H_x &= pI_x + I_p \omega_p \ ; \\ H_y &= qI_y \ ; \qquad \textbf{(2.29)} \\ H_z &= rI_z \ . \end{aligned}$$

Applying equation (2.29) to equation (2.25), the moment equations can be written for the specific case of Archytas as

$$L=I_x\dot{p}+qr(I_z-I_y)+I_p\dot{\omega}_p \;;$$
$$M=I_y\dot{q}+rp(I_x-I_z)+rI_p\omega_p \;; \qquad (2.30)$$
$$N=I_z\dot{r}+pq(I_y-I_x)-qI_p\omega_p \;;$$

and the force equations are those of equation (2.24).

3. Orientation and Position

Because the frame of reference developed for the equations of motion is fixed to the aircraft, and moves with it, the position of the aircraft cannot be described relative to it. The orientation and position of the aircraft can be defined in terms of a fixed frame of reference as shown in Figure 2.3. [Ref. 1: p. 89]

The orientation of the aircraft can be described by a series of three consecutive rotations, whose order is important. The angular rotations are called the *Euler angles*. The orientation of the body frame with respect to the fixed frame can be determined in the following manner. The aircraft is imagined first to be oriented so that its axes are parallel to the fixed frame and the following rotations are then applied. [Ref. 2: pp. 89-91]

1. A rotation Ψ about oz_1, carrying the axes to $Cx_2y_2z_2$ (bringing Cx to its final azimuth).
2. A rotation θ about oy_2, carrying the axes to $Cx_3y_3z_3$ (bringing Cx to its final elevation).
3. A rotation Φ about ox_3, carrying the axes to their final position Cxyz (giving the final angle of bank to the wings).

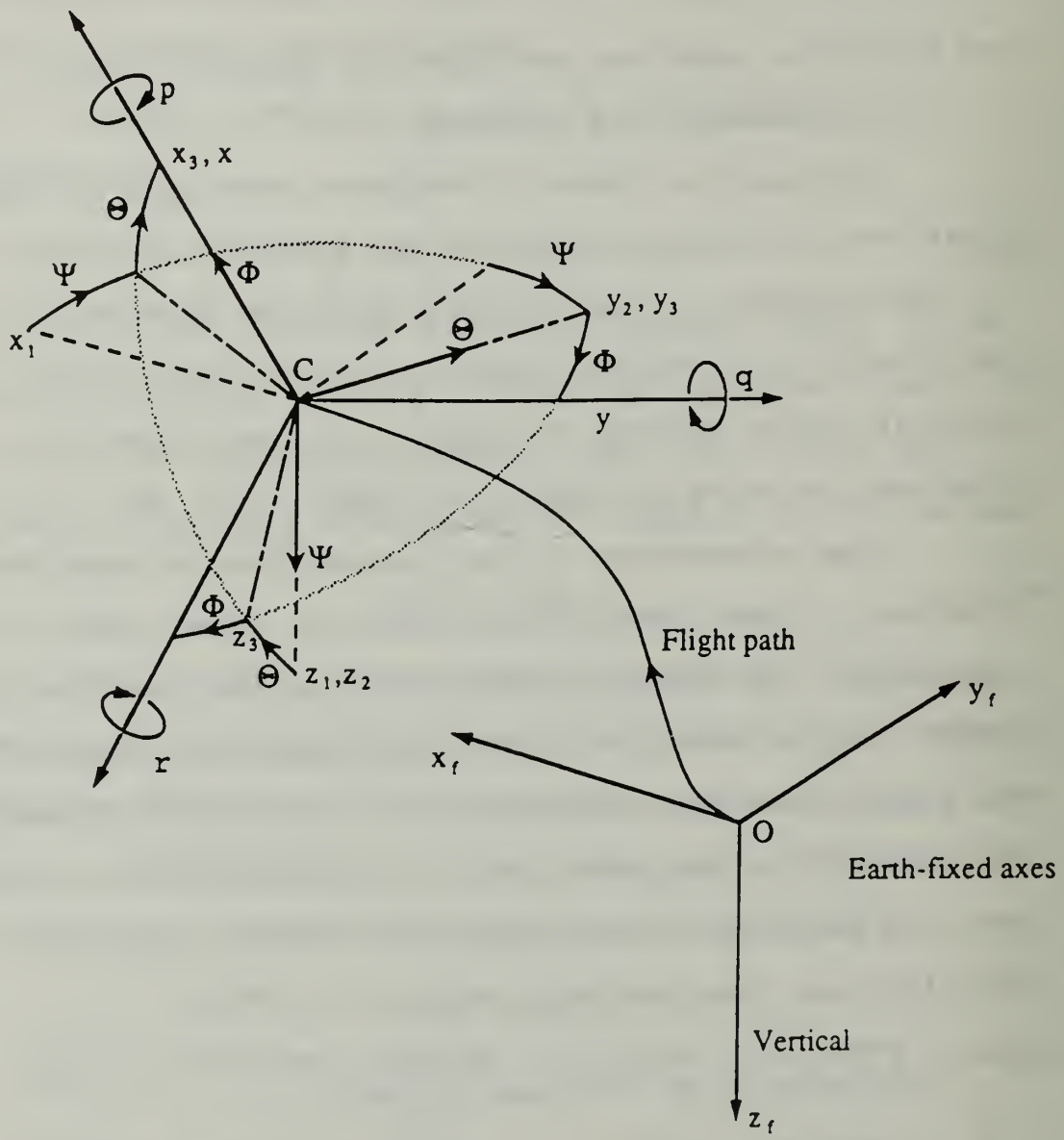


Figure 2.3 Relationship between Body and Fixed Axes System

Now that the *Euler angles* are defined, the flight velocity components relative to the earth-fixed reference frame can be determined. To do this, let the velocity components along the x_1, y_1, z_1 frame be dx/dt, dy/dt, dz/dt and, similarly, let the subscripts 2 and 3 denote the components along x_2, y_2, z_2 and x_3, y_3, z_3, respectively. From Figure 2.3, it can be shown that

$$\frac{dx}{dt}=u_1 \; ; \qquad \frac{dy}{dt}=v_1 \; ; \qquad \frac{dz}{dt}=w_1 \; ; \tag{2.31}$$

where

$$\begin{aligned} u_1 &= u_2\cos\psi - v_2\sin\psi \; ; \\ v_1 &= u_2\sin\psi + v_2\cos\psi \; ; \\ w_1 &= w_2 \; ; \end{aligned} \tag{2.32}$$

and

$$\begin{aligned} u_2 &= u_3\cos\theta + w_3\sin\theta \; ; \\ v_2 &= v_3 \; ; \\ w_2 &= -u_3\sin\theta + w_3\cos\theta \; ; \end{aligned} \tag{2.33}$$

and

$$\begin{aligned} u_3 &= u \; ; \\ v_3 &= v\cos\Phi - \sin\Phi \; ; \\ w_3 &= v\sin\Phi + w\cos\Phi \; ; \end{aligned} \tag{2.34}$$

from this, the absolute velocity in terms of the *Euler angles* and velocity components in the body frame can be

determined. Note the shorthand notation $S_\psi \equiv \sin\Psi$, $C_\psi \equiv \cos\Psi$, $S_\theta \equiv \sin\theta$, etc, used in the following equations:

$$\begin{bmatrix} \frac{dx}{dt} \\ \frac{dy}{dt} \\ \frac{dz}{dt} \end{bmatrix} = \begin{bmatrix} C_\theta C_\psi & S_\Phi S_\theta C_\psi - C_\Phi S_\psi & C_\Phi S_\theta C_\psi + S_\Phi S_\psi \\ C_\theta S_\psi & S_\Phi S_\theta S_\psi + C_\Phi C_\psi & C_\Phi S_\theta S_\psi - S_\theta C_\psi \\ -S_\theta & S_\theta C_\theta & C_\Phi C_\theta \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \qquad (2.35)$$

Integration of these equations yields the aircraft's position relative to the fixed frame of reference. The relationship between the angular velocities in the body frame (p, q and r) and the *Euler rates* ($\dot{\theta}$, $\dot{\psi}$ and $\dot{\Phi}$) can also be determined from Figure 2.3.

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -S_\theta \\ 0 & C_\Phi & C_\theta S_\Phi \\ 0 & -S_\Phi & C_\theta C_\Phi \end{bmatrix} \begin{bmatrix} \dot{\Phi} \\ \dot{\theta} \\ \dot{\Psi} \end{bmatrix} \qquad (2.36)$$

Equations (2.36) can be solved for the *Euler* rates in terms of the body angular velocities and is given by equation (2.37)

$$\begin{bmatrix} \dot{\Phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & s_\Phi \tan\theta & C_\Phi \tan\theta \\ 0 & C_\Phi & -S_\Phi \\ 0 & S_\Phi \sec\theta & C_\Phi \sec\theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \qquad (2.37)$$

By integrating the above equations, the *Euler angles* (θ, ψ and Φ) can be determined.

4. Gravitational and Thrust Forces

The gravitational force acting on the aircraft acts through the center of gravity of the aircraft. Because the body axis system is fixed to the center of gravity, the gravitational force will not produce any moments. However, the gravitational force will contribute to the external force acting on the aircraft and will have components along the respective body axes. From Figure 2.4 the gravitational force component in the direction of each axis is found to be

$$\begin{aligned} X_g &= -mg\cos\theta\cos\Psi \ ; \\ Y_g &= \ mg\cos\theta\sin\Psi \ ; \\ Z_g &= -mg\sin\theta \ . \end{aligned} \tag{2.38}$$

With the aerodynamic forces (including the propulsive forces) denoted by (X, Y, Z), the resultant external forces are

$$\begin{aligned} F_x &= X - mg\cos\theta\cos\Psi ; \\ F_y &= Y + mg\cos\theta\sin\Psi ; \\ F_z &= Z - mg\sin\theta \ . \end{aligned} \tag{2.39}$$

5. Summary of Equations of Motion

In the previous sections, the equations that completely describe the dynamic behavior of Archytas have been developed. Equations (2.24) and (2.30) define the externally applied forces and moments which are represented by F_x, F_y, F_z and L, M, N. Through the *Euler angles*, the

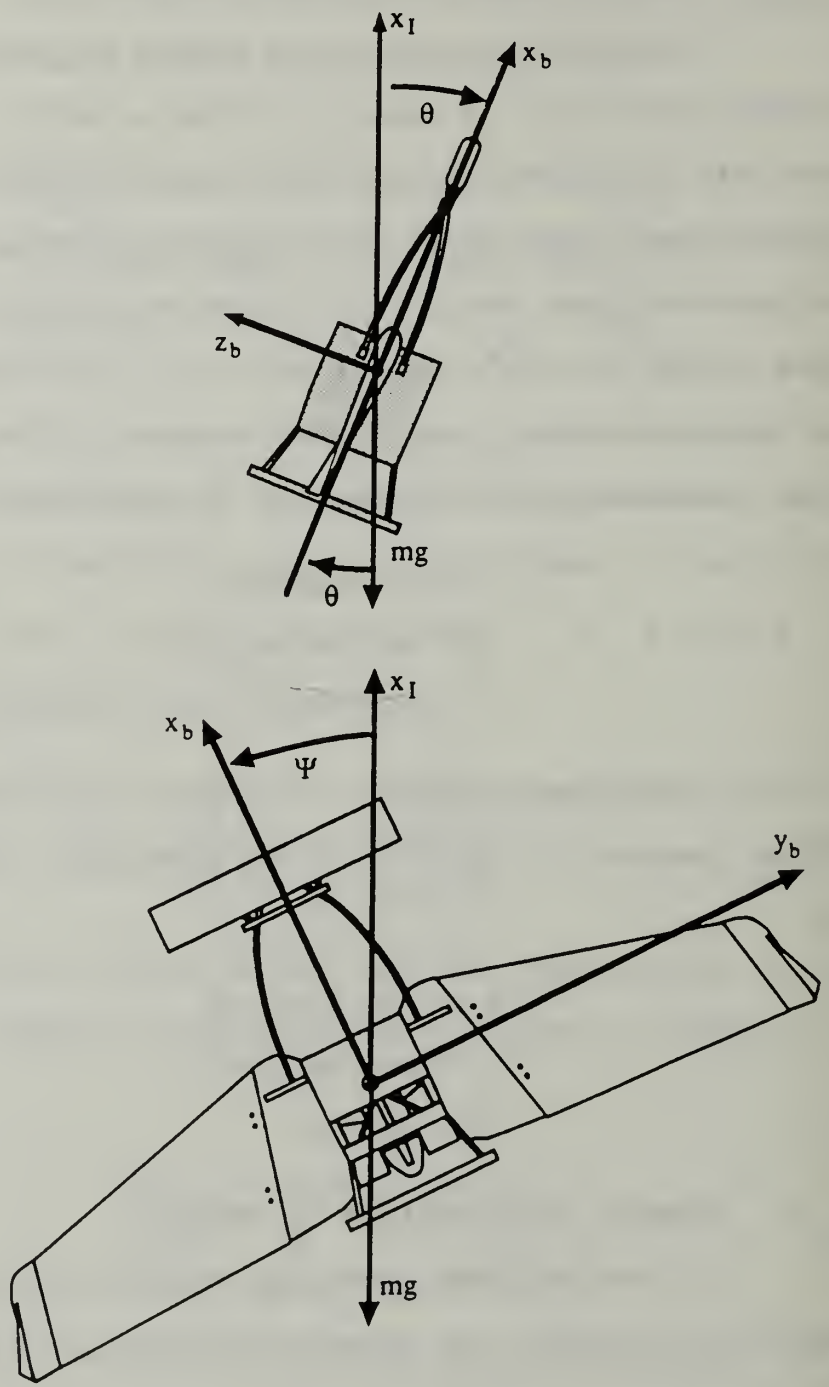


Figure 2.4 Components of Gravity acting along Body Axes

behavior of Archytas can be observed relative to the Earth. Specifically, the translational velocities for the fixed frame of reference, dx/dt, dy/dt and dz/dt, can be determined from the body-fixed velocities, u,v,w and the *Euler angles*, θ, Ψ and Φ, using the transformation of equation (2.35). Additionally, equation (2.37) describes the relationship between the *Euler angles* and the body angular velocities, p, q and r. Table 2.1 gives a summary of the rigid body equations of motion.

B. ARCHYTAS NONLINEAR SYSTEM EQUATIONS

1. Applied Forces and Moments

The Archytas model is developed using the measurements and data from the AROD prototype as first approximations. These measurements and data were obtained by the AROD project engineers based on wind tunnel tests. This data consists of tabular results that describe the aerodynamic lift and drag coefficients and physical measurements of constants such as weight, moments of inertia and servo gains. This tabulated data forms the basis from which the applied forces and moments may be determined. The forces and moments, which are computed from the data are of two types: aerodynamic and thrust. This data is listed in Appendix A.

TABLE 2.1 SUMMARY OF GENERAL EQUATIONS OF MOTION

$$X - mg\,C_\theta C_\Psi = m(\dot{u} + qw - rv)$$
$$Y + mg\,C_\theta S_\Psi = m(\dot{v} + ru - pw) \qquad \text{Force equations}$$
$$Z - mg\,S_\theta = m(\dot{w} + pv - qu)$$

$$L = I_x\dot{p} + qr(I_z - I_y) + I_p\dot{\omega}_p$$
$$M = I_y\dot{q} + rp(I_x - I_z) + rI_p\omega_p \qquad \text{Moment equations}$$
$$N = I_z\dot{r} + pq(I_y - I_x) - qI_p\omega_p$$

$$p = \dot{\Phi} - \dot{\Psi}S_\theta$$
$$q = \dot{\theta}C_\Phi + \dot{\Psi}C_\theta S_\Phi \qquad \text{Angular velocities}$$
$$r = \dot{\Psi}C_\theta C_\Phi - \dot{\theta}S_\Phi$$

$$\dot{\theta} = qC_\Phi - rS_\Phi$$
$$\dot{\Phi} = p + qS_\Phi T_\theta + rC_\Psi T_\theta \qquad \text{Euler rates}$$
$$\dot{\Psi} = (qS_\Phi + rC_\Phi)\sec\theta$$

Velocity of aircraft in the fixed frame in terms of Euler angles and body velocity components

$$\begin{bmatrix} \frac{dx}{dt} \\ \frac{dy}{dt} \\ \frac{dz}{dt} \end{bmatrix} = \begin{bmatrix} C_\theta C_\Psi & S_\Phi S_\theta C_\Psi - C_\Phi S_\Psi & C_\Phi S_\theta C_\Psi + S_\Phi S_\Psi \\ C_\theta S_\Psi & S_\Phi S_\theta S_\Psi + C_\Phi C_\Psi & C_\Phi S_\theta S_\Psi - S_\theta C_\Psi \\ -S_\theta & S_\theta C_\theta & C_\Phi C_\theta \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix}$$

a. Total Angle of Attack and Body Roll Angle

The aerodynamic data describes the forces and moments relative to the vehicle's total velocity vector, v_{TOT}. These forces and moments (in the body-fixed coordinate system) depend on the *total angle of attack* (α') and the *body roll angle* (Δ). The total angle of attack and body roll angle can be defined in terms of the velocity

components as shown in Figure 2.5. The equations for α' and Δ are given as:

$$\alpha' = \sin^{-1}\frac{vwterm}{u} , \tag{2.40}$$

and

$$\Delta = \tan^{-1}\frac{v}{w} . \tag{2.41}$$

b. Aerodynamic Forces and Moments

The forces computed from the tabular data are lift (F_l) and drag (F_d). These forces are depicted in Figure 2.6(a). The transformation from lift and drag to F_{ax}, F_{ay} and F_{az} is given as

$$\begin{bmatrix} F_{ax} \\ F_{ayz} \end{bmatrix} = \begin{bmatrix} \sin\alpha' & -\cos\alpha' \\ -\cos\alpha' & -\sin\alpha' \end{bmatrix} \begin{bmatrix} F_l \\ F_d \end{bmatrix} ; \tag{2.42}$$

and

$$\begin{bmatrix} F_{ay} \\ F_{az} \end{bmatrix} = \begin{bmatrix} \sin\Delta \\ \cos\Delta \end{bmatrix} F_{ayz} \tag{2.43}$$

where F_{ax}, F_{ay} and F_{az} are the forces in the x, y and z directions due to the aerodynamic data.

Similarly, the aerodynamic moments applied to the body axes as a result of the vehicle's movement through the air can be derived. These moments (shown in Figure 2.6(b)) are referred to as the aerodynamic angular moments of roll (L_a), pitch (M_a) and yaw (N_a) and are given as

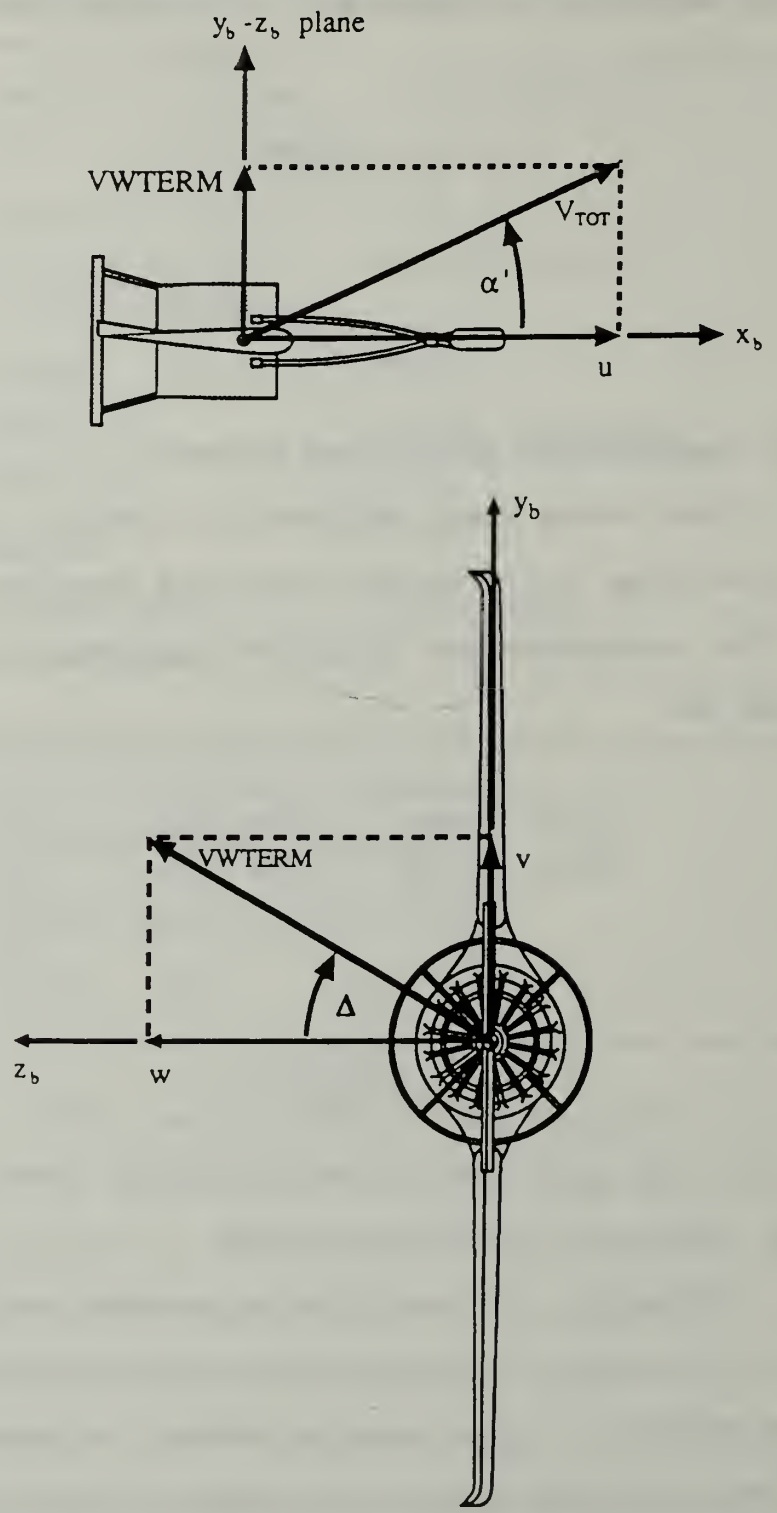
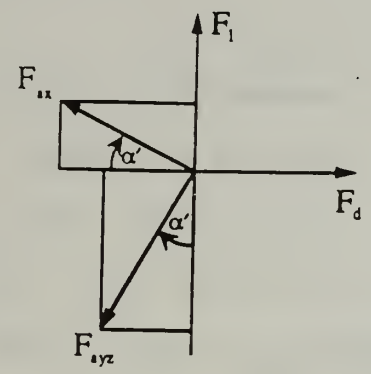
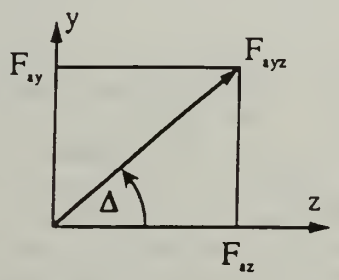


Figure 2.5 Angle of Attack (α′)and Body Roll Angle (Δ)

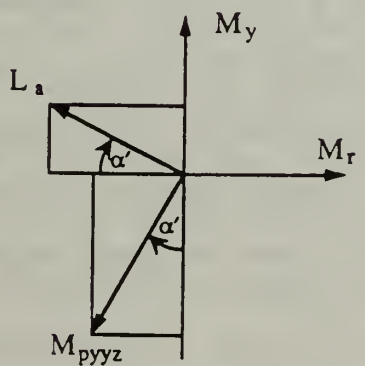


$$\begin{bmatrix} F_{ax} \\ F_{ayz} \end{bmatrix} = \begin{bmatrix} \sin\alpha' & -\cos\alpha' \\ -\cos\alpha' & -\sin\alpha' \end{bmatrix} \begin{bmatrix} F_l \\ F_d \end{bmatrix}$$

$$\begin{bmatrix} F_{ay} \\ F_{az} \end{bmatrix} = \begin{bmatrix} \sin\Delta \\ \cos\Delta \end{bmatrix} F_{ayz}$$

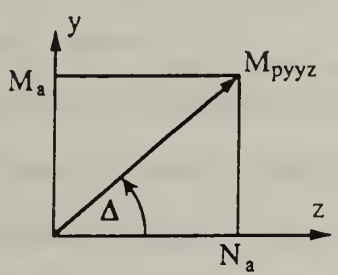


(a) Forces



$$\begin{bmatrix} L_a \\ M_{pyyz} \end{bmatrix} = \begin{bmatrix} \sin\alpha' & -\cos\alpha' \\ -\cos\alpha' & -\sin\alpha' \end{bmatrix} \begin{bmatrix} M_y \\ M_r \end{bmatrix}$$

$$\begin{bmatrix} M_a \\ N_a \end{bmatrix} = \begin{bmatrix} \sin\Delta \\ \cos\Delta \end{bmatrix} M_{pyyz}$$



(b) Moments

Figure 2.6 Aerodynamic Forces and Moments

$$\begin{bmatrix} L_a \\ M_{pyyz} \end{bmatrix} = \begin{bmatrix} \sin\alpha' & -\cos\alpha' \\ -\cos\alpha' & -\sin\alpha' \end{bmatrix} \begin{bmatrix} M_y \\ M_r \end{bmatrix} \quad (2.44)$$

and

$$\begin{bmatrix} M_a \\ N_a \end{bmatrix} = \begin{bmatrix} \sin\Delta \\ \cos\Delta \end{bmatrix} M_{pyyz} \quad (2.45)$$

where M_y, and M_r are the yaw and roll moments relative to V_{TOT}. M_{pyyz} is the moment in the y-z plane. The relationships that define F_l, F_d, and M_y, M_r were developed by the AROD project engineers and are listed in Appendix A. [Ref. 15]

c. Control Forces and Moments Due to Command Inputs

The forces and moments previously discussed are a result of the vehicles motion through the air. A second category includes those forces and moments which are a result of the commanded inputs. These inputs control the rotor speed and the displacement of the control surfaces. [Ref. 4: p. 33]

(1) Forces Due to Induced Thrust of the Ducted Fan. The commanded inputs include the ability to change the rotor speed; thus, changing the thrust provided by the ducted fan. Due to the orientation of the body-fixed axes, the force due to the thrust (F_{Thr}) is directed completely along the x-axis. The relationship that defines F_{Thr} was developed by the AROD project engineers and is given as

$$F_{Thr} = THOVER + XRPM * \delta_{rpm} \quad (2.46)$$

where THOVER is a constant equal to the nominal thrust at hover. THOVER is set equal to Archytas prototype weight of 76.5 ft/lbs. XRPM is the slope of the thrust versus engine rpm curve. For the hover rpm of 712.0943 rad/sec (6800 rpm), XRPM is equal to 0.2387 lb_f/rad/sec. δ_{rpm} is the change in engine rpm from the nominal hover rpm in rad/sec.

(2) Moment Due to Ducted Fan Effects. A gyroscope imparts no torque on its axis if it spins with a constant angular rotation. In hover (constant rotor speed), Archytas behaves similarly to a gyroscope. If the rotor accelerates (positively or negatively) a torque is applied to the axis. This torque is accounted for in Equation (2.30). However, Archytas is a ducted fan and the drag between the rotor tip and the inside body wall creates a moment about the x-axis (roll). The project engineers for AROD determined an approximation for this moment based on experimentation. Because the Archytas duct is identical to the AROD duct, the moment determined for AROD applies to Archytas and is given as

$$L_{Thr} = K_{duct}\delta_{rpm} \quad (2.47)$$

where δ_{rpm} is defined above. K_{duct} is a constant which is dependent on the duct geometry and is equal to 0.0729. [Ref. 4: pp. 33-34]

(3) Moments Due to Control Surface Displacements. The commanded inputs also include the displacement of the four control surfaces (vanes) within the downwash from the duct. The displacement of these vanes within the downwash of the duct imparts moments about the body axes. Figure 2.7 shows how the vanes are symmetrically arranged below the duct. The vanes are displaced by a servo mechanism connected directly to the top of each vane. Vanes (1) and (3) are operated together as "elevators" and impart a moment about the y-axis (pitch). Vanes (2) and (4) together are the "rudder" and contribute a moment about the z-axis (yaw). Vanes (1) and (3) displaced in opposite directions and (2) and (4) displaced oppositely work as "ailerons" to impart a moment about the x-axis (roll). The actual torque applied by each combination of vanes was determined experimentally and described by "constants of effectiveness" which were calculated by the AROD project engineers. These constants of effectiveness can be applied directly to Archytas. [Ref. 4: p.34]

The constants of effectiveness are given the symbols L_{aeff}, M_{eeff} and N_{reff}, for their contribution of moments about the roll, pitch and yaw axes due to a displacement by the ailerons, elevator and rudder. The relationships resulting in moments about the three body axes are

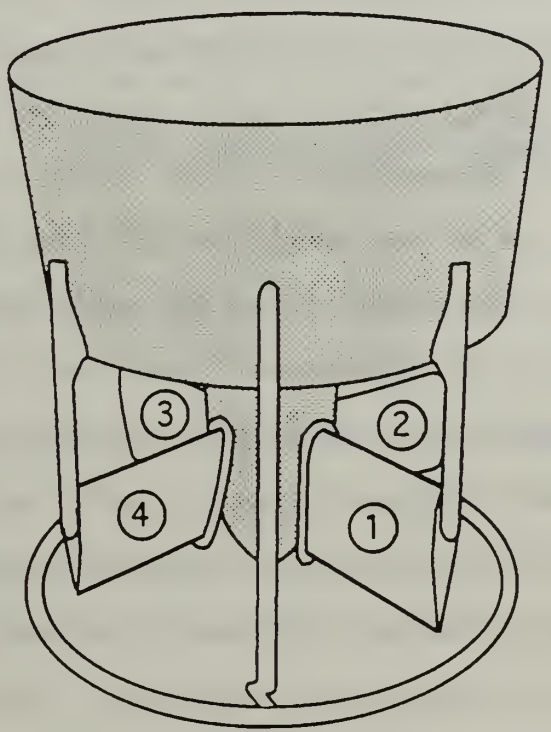


Figure 2.7 Archytas Control Vanes

$$L_t = L_{aeff}\delta_a \ ;$$
$$M_t = M_{eeff}\ P_{factor}\delta_e \ ; \qquad (2.48)$$
$$N_t = N_{reff} Y_{factor}\delta_r \ ;$$

where δ_a, δ_e and δ_r are the displacements of the aileron, elevator and rudder respectively. P_{factor} and Y_{factor} are scaling factors. The relationships that define P_{factor} and Y_{factor} were developed by the AROD project engineers and are listed in Appendix A. L_{aeff}, M_{eeff} and N_{reff} equal -150,379.57, -112,716.87 and -128,774.80, respectively, with units of lb_m-in^2-rad/sec^2. Because of the symmetry of Archytas, cross coupling of the control surfaces is negligible and is ignored. [Ref. 4: pp. 35-36]

2. Servo Equations for Control Surfaces and Throttle

The model airplane servos used to drive the control vanes and throttle linkage are identical and can be modeled as second-order dynamical systems. The response of these servos to a step response was measured. These measurements were used to compute the natural frequency and damping ratio. The results are summarized below in the form of constants H_1 and H_2. A detailed explanation as to how these results were obtained is contained in Appendix B.

a. Control Surface Servos

Three equations will describe the operation of the elevators, rudders and ailerons. For each of these equations, at least two servos are operating at the same

time on different control vanes. Each servo receives a command input (pulse-width modulated signal) which results in an angular displacement of the servo. The corresponding differential equations for the servos can be written as

$$\begin{aligned} \ddot{\delta}_a &= -H_1\dot{\delta}_a - H_2\delta_a + H_2 u_a \ ; \\ \ddot{\delta}_e &= -H_1\dot{\delta}_e - H_2\delta_e + H_2 u_e \ ; \\ \ddot{\delta}_r &= -H_1\dot{\delta}_r - H_2\delta_r + H_2 u_r \ ; \end{aligned} \tag{2.49}$$

where H_1 and H_2 are the servo gain constants equal to 71.1 and 2745.8. u_a, u_e and u_r are the servo inputs. δ_a, δ_e and δ_r are the servo position angles.

b. Throttle Servos and Engine

The servo motor used to open and close the throttle is identical to the servo motors used for the control surfaces. The 2-cycle, 2-cylinder gasoline engine with dual carburetors can be modeled as a first order lag system. The complete third order system is

$$\begin{aligned} \ddot{\delta}_t &= -H_1\dot{\delta}_t - H_2\delta_t + H_2 u_t \ ; \\ \dot{\delta}_{rpm} &= -\omega_E\delta_{rpm} + \omega_E K_E \delta_T \ ; \end{aligned} \tag{2.50}$$

where ω_E is the lag time constant equal to 2.0 rad/sec and K_E is a scaling factor equal to 837.758 rad/sec/rad.

3. Summary

The result of the previous section was nine equations completely describing the dynamic behavior of Archytas. These equations, combined with the equations that

define the servos and engine, form the complete nonlinear system. This nonlinear system will be the base model throughout this thesis. Table 2.2 lists the equations rearranged so that the dynamic variables of interest may be determined. Additionally, the applied forces and moments defined in this section have been substituted into the equations. These applied forces and moments complete the development of the model for the specific case of Archytas.

C. LINEARIZING THE ARCHYTAS MODEL

There exist many analytical and graphical techniques for controller design and analysis of linear systems. Conversely, there are no good methods available for solving a wide class of nonlinear systems. Thus, in the design of control systems it is practical to first design the controller based on the linear system model generated by neglecting the nonlinearities of the system. These nonlinearities are neglected by linearizing the model about a steady-state reference condition. The designed controller is then applied to the nonlinear system model for validation and subsequent redesign if necessary. In this section, a linear model is generated from the equations of Table 2.2 based on steady-state assumptions and physical approximations. [Ref. 5: p. 11]

The nonlinearities of the Archytas system equations fall into two categories: (1) nonlinear combination of states

Table 2.2 ARCHYTAS NONLINEAR SYSTEM EQUATIONS

$$\dot{p}=\frac{1}{I_x}\left[L_a(M_y,M_z)+L_{Thr}(\delta_{rpm})+L_t(\delta_a)+(I_y-I_z)rq-I_{rx}\dot{\omega}_p\right]$$

$$\dot{q}=\frac{1}{I_y}\left[M_a(M_y,M_z)+M_t(\delta_e)+(I_z-I_x)pr-I_{rx}\omega_p r\right]$$

$$\dot{r}=\frac{1}{I_z}\left[N_A(M_y,M_z)+N_t(\delta_r)+(I_x-I_Y)pq+I_{rx}\omega_p q\right]$$

$$\dot{u}=(rv-qw)-g\cos\theta\cos\Psi+\frac{1}{m}\left[F_{ax}(F_l,F_d)+F_{Thrx}(\delta_{rpm})\right]$$

$$\dot{v}=(pw-ru)+g\cos\theta\sin\Psi+\frac{1}{m}F_{ay}(F_l,F_d)$$

$$\dot{w}=(qu-pv)-g\sin\theta+\frac{1}{m}F_{az}(F_l,F_d)$$

$$\dot{\theta}=q\cos\Phi-r\sin\Phi$$

$$\dot{\Phi}=p+q\sin\Phi\tan\theta+r\cos\Phi\tan\theta$$

$$\dot{\Psi}=(q\sin\Phi+r\cos\Phi)\sec\theta$$

$$\ddot{\delta}_a=-H_1\dot{\delta}_a-H_2\delta_a+H_2u_a$$

$$\ddot{\delta}_e=-H_1\dot{\delta}_e-H_2\delta_e+H_2u_e$$

$$\ddot{\delta}_r=-H_1\dot{\delta}_r-H_2\delta_r+H_2u_r$$

$$\ddot{\delta}_t=-H_1\dot{\delta}_t-H_2\delta_t+H_2u_t$$

$$\dot{\delta}_{rpm}=-\omega_E\delta_{rpm}+\omega_E K_E\delta_t$$

(e.g. Equation (2.30)) and (2) discontinuous functions (e.g. table lookup of aerodynamic force and moment coefficients). These nonlinearities will be neglected and the nonlinear equations will be replaced with linear approximations.

1. Steady-State Assumptions

The motion of Archytas in the hover mode consists of small perturbations from a steady-state condition. The

steady-state hover condition is defined as α' equal to 90°. All translational and angular movement is very small. The steady-state hover condition results in the following simplifications of the nonlinear system equations.

1. The product of two small numbers is an extremely small number, thus terms involving the products of translational or angular velocities are equal to zero (e.g. rq, pq, (pw-ru) are set equal to zero).

2. The aerodynamic forces (F_{ax}, F_{ay}, F_{az}) and moments (L_a, M_a, N_a) are very small for α' equal to 90°, and can be neglected in the hover flight condition. (Note: Because the vanes lie within the downwash of the propeller, the vane effectiveness coefficients can not be neglected.)

3. The sine of a state is equal to the state and the cosine of a state is equal to one. This is the small angle approximation for angles less than 15 degrees.

4. K_{duct} is a small number equal to 0.0729. For hover or very small translational velocities, δ_{rpm} is a very small number. Therefore, the product of K_{duct} and δ_{rpm} is neglected.

5. The propeller angular velocity, ω_p, is considered constant. Thus, the propeller angular acceleration, $\dot{\omega}_p$, is equal to zero.

Table 2.3 lists the Archytas system equations when the above simplifications are applied to the nonlinear system equations. Note that much of the coupling between states and all of the nonlinear products of states have been eliminated.

Table 2.3 ARCHYTAS LINEARIZED HOVER EQUATIONS

$$\dot{p}=L_t(\delta_a)$$

$$\dot{q}=M_t(\delta_e)-\frac{I_{rx}\omega_p}{I_x}r$$

$$\dot{r}=N_t(\delta_r)+\frac{I_{rx}\omega_p}{I_z}q$$

$$\dot{u}=\frac{F_{Thrx}}{m}-g$$

$$\dot{v}=g\Psi$$

$$\dot{w}=-g\theta$$

$$\dot{\theta}=q$$

$$\dot{\Phi}=p$$

$$\dot{\Psi}=r$$

$$\ddot{\delta}_a=-H_1\dot{\delta}_a-H_2\delta_a+H_2u_a$$

$$\ddot{\delta}_e=-H_1\dot{\delta}_e-H_2\delta_e+H_2u_e$$

$$\ddot{\delta}_r=-H_1\dot{\delta}_r-H_2\delta_r+H_2u_r$$

$$\ddot{\delta}_t=-H_1\dot{\delta}_t-H_2\delta_t+H_2u_t$$

$$\dot{\delta}_{rpm}=-\omega_E\delta_{rpm}+\omega_E K_E\delta_t$$

2. Physical Approximations of Force and Moments

The force, F_{Thrx}, in Table 2.3 is a function of the engine rpm and is computed by equation (2.46). The moment terms, L_t, M_t and N_t, are functions of the displacement of the aileron, elevator and rudder. They are computed by equation (2.48). The moments of inertia, I_x, I_y, I_z and I_{rx}, were determined by the AROD project engineers and are listed in Appendix A. The angular velocity of the propeller, ω_p, in

the hover mode equals 712.0943 rad/sec (6800 rev/min). The weight of the AROD prototype is equal to 76.5 lbs. Using these constant parameters, the forces and moments for the hover steady state reference condition were computed and are listed in Table 2.4. Substituting the results of Table 2.4 into the equations of Table 2.3 yields the linearized hover equations, which are summarized in Table 2.5.

D. SUMMARY

The nonlinear differential equations of motion of a rigid body were developed from Newton's second law. The equations were then modified for the specific case of Archytas. This modification included the effect of the spinning three-bladed rotor. The equations were linearized about a steady-state hover condition. Next, the applied forces and moments, and the servo and engine equations were defined. The forces and moments specific for the hover reference condition were computed and applied to the Archytas linear model.

Table 2.4 FORCES, MOMENTS AND CONSTANTS

Moments

$L_t = L_{aeff}\delta_a$

$L_T = -150,379.57\delta_a$

$M_t = M_{eeff}P_{factor}\delta_e$

$M_t = (-112,716.87)(1)\delta_e$

$N_t = N_{reff}Y_{factor}\delta_r$

$N_t = (-128,774.80)(1)\delta_r$

Forces

$F_{Thrx} = THOVER + XRPM\ \delta_{rpm}$

$F_{Thr} = 76.5 + 0.2387\delta_{rpm}$

Moments of Inertia

$I_x = 7063.39$, $I_y = 7768.22$, $I_z = 7729.58$, $I_{rx} = 69.552$ $(lb_m\ in^2)$

Angular velocity

$\omega_p = 712.0943 \frac{rad}{sec}$;

Weight/gravity/mass

$weight = 76.5\ lbs.$; $gravity = 32.174 \frac{ft}{sec^2}$; $mass = \frac{weight}{gravity}$

Engine Lag Time Constant and Scale Factor

$\omega_E = 2.0 \frac{rad}{sec}$, $K_E = 837.758\ rad/sec/rad$

Table 2.5 LINEARIZED HOVER STATE EQUATIONS

STATE

$$\dot{p}=\frac{-150,379.57}{7063.39}\delta_a=-21.29\delta_a$$

$$\dot{q}=\frac{1}{7768.22}[-112,716.87\delta_e-52,440.82r]=-14.51\delta_e-6.75r$$

$$\dot{r}=\frac{1}{7729.58}[-128,774.80\delta_r+52,440.82q]=-16.68\delta_r+6.78q$$

$$\dot{u}=\frac{(76.5+0.2387\delta_{rpm})}{2.38}-32.174=0.10029\delta_{rpm}$$

$$\dot{v}=32.174\Psi$$

$$\dot{w}=-32.174\theta$$

$$\theta=q$$

$$\Phi=p$$

$$\Psi=r$$

$$\ddot{\delta}_a=-H_1\dot{\delta}_a-H_2\delta_a+H_2u_a$$

$$\ddot{\delta}_e=-H_1\dot{\delta}_e-H_2\delta_e+H_2u_e$$

$$\ddot{\delta}_r=-H_1\dot{\delta}_r-H_2\delta_r+H_2u_r$$

$$\ddot{\delta}_t=-H_1\dot{\delta}_t-H_2\delta_t+H_2u_t$$

$$\dot{\delta}_{rpm}=-\omega_E\delta_{rpm}+\omega_EK_E\delta_t$$

III. OPTIMAL CONTROL THEORY

The purpose of this chapter is to develop a procedure for applying optimal control theory to the solution of tracking problems. First, several reasons for desiring an optimal solution are presented. Next, the state space representation (both continuous and discrete) of a system is given. Finally, the application of optimal control to the solution of tracking problems is illustrated.

A. WHY OPTIMUM CONTROL FOR ARCHYTAS?

The first reason for seeking an optimum controller is that feedback gains can be computed for a much broader range of control problems. Specifically, optimal control provides solutions for high order, multiple-input, multiple-output (MIMO) systems. Such systems are often intractable with classical methods. The pitch and yaw angle controller for Archytas is an eight order MIMO system. Thus, optimal control is the preferred method.

Additionally, optimal control lends itself nicely to a discrete time solution of the control problem. Archytas will employ an on board digital computer to perform inflight stability and control. While a continuous time controller can be easily discretized in many cases, design of a sampled data controller will simplify the procedure.[Ref. 3: p. 58]

Finally, optimal control is known to provide robust and insensitive solutions to the feedback control problem. Assuming that an appropriate performance measure is chosen to determine the optimal feedback gain matrix, K, the solution can be expected to have a fair degree of tolerance to plant model inaccuracies. Clearly, robustness is not only a desired property of the controller, it is an absolute necessity if the controller is to be applicable to both the linear and nonlinear models of Archytas.[Ref. 3: pp. 58-59]

B. STATE SPACE REPRESENTATION

The *state* of a system may be defined as the minimum amount of information required such that (given the input to the system) the response of the system is completely determined for all future time. For dynamic systems, the response of the system is defined by the differential equations that model the system, the initial conditions and the forcing function. The number of state variables or *states* is equal to the total order of the systems differential equations. In order to provide a systematic mathematical approach to analysis of the system, it is convenient to describe the system by a set of first-order differential equations. This set of equations is called the *state equations* and constitute the basis for the state space representation. [Ref. 8: pp. 206-207]

1. Continuous Time Systems

The state space representation of a general n^{th} order continuous-time, time-invariant system is described by the following matrix state equations:

$$\dot{\underline{x}}(t) = \underline{\underline{A}}\,\underline{x}(t) + \underline{\underline{B}}\,\underline{u}(t) \; ; \qquad (3.1)$$

$$\underline{y}(t) = \underline{\underline{C}}\,\underline{x}(t) \; ; \qquad (3.2)$$

where the definitions in Table 3.1 apply to a system with ℓ control inputs and m measurable outputs. The *equations*

TABLE 3.1
STATE SPACE DEFINITIONS FOR CONTINUOUS-TIME SYSTEMS

Term	Dimension	Definition
$\underline{x}(t)$	$(n \times 1)$	State vector
$\underline{y}(t)$	$(m \times 1)$	Output vector
$\underline{\underline{A}}$	$(n \times n)$	Plant matrix
$\underline{\underline{B}}$	$(n \times \ell)$	Control distribution matrix
$\underline{\underline{C}}$	$(m \times n)$	Output Distribution matrix

listed in Table 2.5 are linear and time-invariant.

2. Discrete Time Systems

As was noted earlier, there are many benefits for seeking a discrete time solution for the Archytas control problem. Therefore, the automatic control systems designed will focus on the application of optimal control theory to discrete time systems.

Similar to the continuous-time system, the state space representation of a general n^{th} order discrete-time system is described by the following matrix state equations:

$$\underline{x}(k+1) = \underline{\underline{\Phi}}\,\underline{x}(k) + \underline{\underline{\Gamma}}\,\underline{u}(k) \quad ; \tag{3.3}$$

$$\underline{y}(k) = \underline{\underline{C}}\,\underline{x}(k) \quad ; \tag{3.4}$$

$\underline{\underline{\Phi}}$ and $\underline{\underline{\Gamma}}$ are defined as:

$$\underline{\underline{\Phi}} = e^{\underline{\underline{A}}t} \quad ; \tag{3.5}$$

$$\underline{\underline{\Gamma}} = \int_0^T e^{\underline{\underline{A}}t}dt\ \underline{\underline{B}} \quad ; \tag{3.6}$$

where T is the sampling period and k is an integer time index. Reference 10 provides a detailed development of the relationship between continuous-time and discrete-time systems. The definition in Table 3.2 apply to a system with ℓ control inputs and m measurable outputs.

TABLE 3.2
STATE SPACE DEFINITIONS FOR DISCRETE-TIME SYSTEMS

Term	Dimension	Definition	
$\underline{x}(k)$	$(n \times 1)$	State vector	
$\underline{y}(k)$	$(m \times 1)$	Output vector	$(0 < m \leq n)$
$\underline{\underline{\Phi}}$	$(n \times n)$	Plant matrix	
$\underline{\underline{\Gamma}}$	$(n \times \ell)$	Control distribution matrix	
$\underline{\underline{C}}$	$(m \times n)$	Output Distribution matrix	

C. LINEAR QUADRATIC REGULATOR PROCEDURE

The theory behind the linear quadratic regulator is well developed. Procedures exist which make the design of controllers using linear quadratic regulator theory easily obtainable. The purpose of this section is to illustrate the application of the linear quadratic regulator to the tracking problems associated with Archytas. Reference 11 provides a detailed development of the linear quadratic regulator theory.

1. Quadratic Cost Function

The discrete LQR synthesis problem is that of determining the control that minimizes the performance measure:

$$J=\sum_{k=0}^{\infty} x^T(k)\,Q\,x(k) + u^T(k)\,R\,u(k)\ ; \qquad (3.7)$$

where

J = Scalar cost of operating the system;

x(k) = State vector at discrete times;

$\underline{u}(k)$ = Control vector at discrete times;

$\underline{\underline{Q}}$ = Symmetric state trajectory weighting matrix;

$\underline{\underline{R}}$ = Symmetric control weighting matrix;

T = Matrix transpose operator.

The solution to this optimization problem is the linear controller:

$$u(k) = -(\underline{\underline{R}}^{-1}\underline{\underline{B}}^T \underline{\underline{M}})\underline{x}(k) = -\underline{\underline{K}}\,\underline{x}(k) \; ; \qquad (3.8)$$

where $\underline{\underline{M}}$ satisfies the *algebraic Riccati equation* (ARE):

$$\underline{\underline{Q}} = \underline{\underline{M}}\,\underline{\underline{A}} + \underline{\underline{A}}^T\underline{\underline{M}} - \underline{\underline{M}}\,\underline{\underline{B}}\,\underline{\underline{R}}^{-1}\underline{\underline{B}}^T\underline{\underline{M}} + \underline{\underline{Q}} \; . \qquad (3.9)$$

[Ref. 11: pp. 345-346] Many software packages, including the program MATLAB used in this thesis, contain subroutines that calculate the value of $\underline{\underline{K}}$ for a given dynamic system and performance measure.

2. Performance Weighting Matrices

The optimal control is fully specified by the system model and the weighting matrices $\underline{\underline{Q}}$ and $\underline{\underline{R}}$. $\underline{\underline{Q}}$ penalizes deviation of the state vector $\underline{x}$ from the origin and $\underline{\underline{R}}$ penalizes the use of too much control effort. Trial and error was used in selecting values for these performance weighting matrices. The relationship between the weighting matrices $\underline{\underline{Q}}$ and $\underline{\underline{R}}$ and the dynamic behavior of the closed-loop system depends of course on the matrices $\underline{\underline{A}}$ and $\underline{\underline{B}}$ and is quite complex. The approach taken in the design of the controllers for Archytas was to solve for the gain matrices $\underline{\underline{K}}$ that result from a range of weighting matrices $\underline{\underline{Q}}$ and $\underline{\underline{R}}$, and then simulate the corresponding closed-loop response. The gain matrix $\underline{\underline{K}}$ that produced the response closest to the desired design objectives was chosen. [Ref. 11: p. 341]

3. Optimal Tracking Systems

The regulator and tracking problems appear very similar. The tracking problem attempts to drive the states of the system to a desired level; whereas, the regulator attempts to drive all of the states to zero. Their differences present conceptual difficulties and sensitivity problems when viewed in a practical context [Ref. 12: pp. 643-647]. Burl [Ref. 13] presents a comprehensive development of the subtleties encountered when applying LQR synthesis to the tracking problem. This development is generalized below to demonstrate these subtleties and to indicate a procedure for selecting the proper form of the performance measure. The following development is applied to a first order system, but the results are applicable to systems of arbitrary order.

Given the scalar plant:

$$\dot{y}(t) = -Ay(t) + Bu(t) \quad . \tag{3.10}$$

The purpose of the control system is to drive the output y(t) to the constant reference input r. This results in the error equation:

$$e(t) = r - y(t) \quad ; \tag{3.11}$$

where the desire is to drive e(t) to zero. The application of linear regulator theory requires that this error be a linear combination of the states of the system. This can be

readily accomplished by generating a new state equation for e(t):

$$\dot{e}(t) = -\dot{y}(t) = Ay(t) - Bu(t) \ ; \tag{3.12}$$

adding and subtracting Ar yields:

$$\begin{aligned} \dot{e}(t) &= Ay(t) + Ar - Ar - Bu(t) \ ; \\ \dot{e}(t) &= A[-r + y(t)] - Bu(t) + Ar \ ; \\ \dot{e}(t) &= -Ae(t) - Bu(t) + Ar \ . \end{aligned} \tag{3.13}$$

Since the error equation, Equation (3.19), is linear and time invariant, the performance measure:

$$J = \int_0^\infty \{Qe^2(t) + u^2(t)\}\, dt \ ; \tag{3.14}$$

should provide an optimal solution for the system of Equation (3.21). However, as shown by Burl [Ref. 13] this optimal control problem will not yield a solution. To demonstrate this fact, note that the existence of this integral requires that both u(t) and e(t) approach zero as t tends to infinity. If e(t) approaches zero, then $\dot{e}(t)$ must also approach zero which from Equation (3.21) yields:

$$0 = -A(0) - Bu(t) + Ar \rightarrow u(t) = \frac{A}{B} r \neq 0 \tag{3.15}$$

This nonzero value for u(t) implies that the performance measure is infinite. A solution to this problem would be to formulate the performance measure as

$$J=\int_0^\infty [Qe^2(t)+\{u(t)-\frac{A}{B}r\}^2]\,dt \ . \qquad (3.16)$$

This will result in the existence of a theoretically optimal control provided that the model of the system (the parameters A and B) is exactly known. When errors exist in the model, a nonzero steady state error will exist. The resulting control system will be mathematically correct, but it will be unacceptable for many applications due to its sensitivity to changes in the plant parameters. The control system will be *super-tuned* (it will not be robust).[Ref. 12: p. 645]

This undesirable result can be overcome by letting the controller find the appropriate steady state value of the control. This is achieved by application of the performance measure:

$$J=\int_0^\infty [Qe^2(t)+\dot{u}^2(t)]\,dt \ . \qquad (3.17)$$

The control that minimizes Equation (3.25) given the system of Equation (3.21) is found by first differentiating Equation (3.21), yielding:

$$\ddot{e}(t)=-A\dot{e}(t)-B\dot{u}(t) \ . \qquad (3.18)$$

Noting that the error is the integral of $\dot{e}(t)$ results in the state space system:

$$\begin{bmatrix}\dot{e}(t)\\ \ddot{e}(t)\end{bmatrix}=\begin{bmatrix}0 & 1\\ 0 & -A\end{bmatrix}\begin{bmatrix}e(t)\\ \dot{e}(t)\end{bmatrix}+\begin{bmatrix}0\\ -B\end{bmatrix}\dot{u}(t) \qquad (3.19)$$

This system and the performance measure of Equation (3.25) form a linear regulator problem whose solution is state feedback:

$$\dot{u}(t)=-[k_1 \quad k_2]\begin{bmatrix}e(t)\\ \dot{e}(t)\end{bmatrix}. \qquad (3.20)$$

The gains k_1 and k_2 are computed by application of linear regulator theory. The actual control that is applied to the system is found by integrating Equation (3.28):

$$u(t)=-k_1\int_0^t e(\tau)\,d\tau-k_2 e(t) \qquad (3.21)$$

The resulting controller incorporates proportional plus integral feedback of the error. This fact is reasonable since the system to be controlled, Equation (3.21), is of type 0 with a steady state disturbance input.

To summarize, when LQR synthesis is applied to the tracking problem, the proper choice of state variables will help to eliminate sensitivity problems and ensure system robustness. The state variables should not contain any input dependent terms, and they should asymptotically approach zero. The system tracking error and its time derivative are natural state variable for the LQR synthesis procedure. In essence, the key to a successful design

depends upon the proper formulation of the performance measure. The performance measure of Equation (3.25) represents one possible formulation that yields reliable results. This performance measure is used in the design of the Archytas control systems. These ideas are extended to a general multiple input, multiple output (MIMO) system by Burl [Ref. 13].

D. SUMMARY

In this chapter, the application of the linear quadratic regulator to the tracking problem has been addressed. A procedure for formulating a performance measure applicable to tracking problems was developed. Additionally, the state space representations for both continuous-time and discrete-time systems were presented. In the next chapter, the results of this chapter are applied to generate control systems for Archytas.

IV. CONTROL SYSTEM DESIGN FOR ARCHYTAS

The purpose of this chapter is to use optimal control theory to design an automatic flight control system for Archytas during the hover mode of flight. Because of the coupled nature of the Archytas control problem, a linear quadratic regulator (LQR) synthesis technique is pursued.

A. ARCHYTAS CONTROL SUBSYSTEMS

The automatic control system is logically separated into three subsystems according to the linearized equations of Table 2.5. The three control subsystems are:

1. Roll rate controller;
2. Altitude rate controller;
3. Pitch angle and yaw angle controller.

Because each of these control subsystems is designed independently, any cross-coupling which may occur between the subsystems is not taken into account.

B. ARCHYTAS ROLL RATE CONTROLLER

1. The Roll System

The roll rate controller for Archytas will serve three primary functions:

1. Allow the operator to command a desired rotation velocity about the vehicle's longitudinal, or x, axis. This rotation velocity will permit the operator to position Archytas by terminating the rolling motion when a desired angle is achieved. This capability is critical during the landing phase in order to position Archytas correctly with respect to the wind.

2. Eliminate the rotation velocity imparted to Archytas from the effect of cross-winds.

3. Eliminate the rotation velocity imparted to Archytas from the reactive torques applied to the roll axis as the engine speed is varied.

The simplified equation of motion which describes Archytas' roll rate subsystem is given as:

$$\dot{p}=-21.29\delta_a \ . \qquad \textbf{(4.1)}$$

The aileron servo dynamics are modeled in Appendix B as a second order dynamical system with a natural frequency, ω_n, of 52.4 rad/sec (8.34 Hz) and a damping ratio, ζ, of 0.707. The corresponding differential equation is:

$$\ddot{\delta}_a=-74.1\dot{\delta}_a-2745.8\delta_a+2745.8u_a \ . \qquad \textbf{(4.2)}$$

From Equations (4.1) and (4.2), the state equation can be written in the matrix form $\dot{\underline{x}}=\underline{\underline{A}}\,\underline{x}+\underline{\underline{B}}\,u$:

$$\begin{bmatrix} \dot{p} \\ \dot{\delta}_a \\ \ddot{\delta}_a \end{bmatrix} = \begin{bmatrix} 0 & -21.29 & 0 \\ 0 & 0 & 1 \\ 0 & -2745.8 & -71.1 \end{bmatrix} \begin{bmatrix} p \\ \delta_a \\ \dot{\delta}_a \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 2745.8 \end{bmatrix} u_a \ . \qquad \textbf{(4.3)}$$

The roll system tracking error is defined as:

$$E_p = p_c - p \ ; \tag{4.4}$$

where p_c is the input command and p is the measured roll rate. From Equation (4.4), the differential equation for the tracking error is:

$$\dot{E}_p = -\dot{p} \ . \tag{4.5}$$

If Equation (4.5) is combined with the time derivatives of Equations (4.1) and (4.2) a new state equation can be formed that is appropriate for tracking system design (as discussed in Chapter III). This state equation is:

$$\begin{bmatrix} \dot{E}_p \\ \ddot{p} \\ \dot{\delta}_a \\ \ddot{\delta}_a \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 0 & 0 & -21.29 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -2745.8 & -74.1 \end{bmatrix} \begin{bmatrix} E_p \\ \dot{p} \\ \dot{\delta}_a \\ \ddot{\delta}_a \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 2745.8 \end{bmatrix} \dot{u}_a \ . \tag{4.6}$$

2. Roll Rate Controller Design

a. Sampling Frequency Selection

The application of optimal control theory to discrete-time systems requires that an appropriate sampling frequency be determined. The sampling frequency, f_s, is simply the inverse of the sampling period, T. A general rule of thumb in control systems is to sample a system such that

$$f_s \geq 10 f_{max} \; ; \qquad (4.7)$$

where f_{max} is the maximum bandwidth of the system. From Appendix B, the natural frequency of the servos is 52.4 rad/sec (8.34 Hz). Bassett [Ref. 3: p. 76] demonstrates how the Archytas MIMO system can be considered as a set of several SISO systems. Each system has a different bandwidth for the purpose of determining a sampling frequency. The subsequent highest natural frequency is equal to 4.64 rad/sec (0.74 Hz). Because the natural frequency of the servos is a factor of ten greater than the highest natural frequency of 4.64 rad/sec, the servo natural frequency will be used to determine the required sampling period, T. From Equation (4.7), f_s=10(52.4 rad/sec)=524.0 rad/sec (83.4 Hz), using 83.4 Hz as the sampling rate yields

$$T = \frac{1}{f_s} = \frac{1}{83.4} = 0.012 \text{ sec} \; . \qquad (4.8)$$

For the controller designs of this thesis, a sampling period of 0.01 seconds will be used.

b. Discretizing the Roll Rate System

MATLAB provides tools for computing the discrete-time matrix state equation. With the sampling period T equal to 0.01 seconds, the discrete-time state space can be written in the matrix form $\underline{x}(k+1) = \underline{\underline{\Phi}}\, \underline{x}(k) + \underline{\underline{\Gamma}}\, \underline{u}(k)$:

$$\begin{bmatrix} E_p(k+1) \\ \dot{p}(k+1) \\ \delta_a(k+1) \\ \dot{\delta}_a(k+1) \end{bmatrix} = \begin{bmatrix} 1 & -0.01 & 0.001 & 0 \\ 0 & 1 & -0.2048 & -0.0008 \\ 0 & 0 & 0.8935 & 0.0067 \\ 0 & 0 & -18.5258 & 0.3936 \end{bmatrix} \begin{bmatrix} E_p(k) \\ \dot{p}(k) \\ \delta_a(k) \\ \dot{\delta}_a(k) \end{bmatrix} + \begin{bmatrix} 0 \\ -0.0081 \\ 0.1065 \\ 18.5258 \end{bmatrix} \dot{u}_a(k) . \quad \textbf{(4.9)}$$

c. Gain Determination

The optimal control is determined from the state equation and the performance measure:

$$J = \sum_{k=0}^{\infty} \left\{ \begin{bmatrix} E_p(k) & \dot{p}(k) & \delta_a(k) & \dot{\delta}_a(k) \end{bmatrix} \underline{\underline{Q}} \begin{bmatrix} E_p(k) \\ \dot{p}(k) \\ \delta_a(k) \\ \dot{\delta}_a(k) \end{bmatrix} + \dot{u}^T(k) \underline{\underline{R}}\, \dot{u}(k) \right\} . \quad \textbf{(4.10)}$$

The optimal control $\dot{u}(k)$ is:

$$\dot{u}(k) = - \begin{bmatrix} k_1 & k_2 & k_3 & k_4 \end{bmatrix} \begin{bmatrix} E_p(k) \\ \dot{p}(k) \\ \delta_a(k) \\ \dot{\delta}_a(k) \end{bmatrix} = - \underline{\underline{K}}\, \underline{e}(k) \quad \textbf{(4.11)}$$

where $\underline{\underline{K}}$ is the steady-state gain matrix. The actual control u(k) is then obtained by integrating $\dot{u}(k)$ to obtain:

$$u(k) = \sum_{m=0}^{k} \dot{u}(m) = - \sum_{m=0}^{k} \begin{bmatrix} k_1 & k_2 & k_3 & k_4 \end{bmatrix} \begin{bmatrix} E_p(m) \\ \dot{p}(m) \\ \delta_a(m) \\ \dot{\delta}_a(m) \end{bmatrix} = \quad \textbf{(4.12)}$$

$$-k_1 \sum_{m=0}^{k} E_p(m) - k_2 p(m) - K_3 \delta_a(m) - k_4 \dot{\delta}_a(m)$$

The LQR weighting matrices, $\underline{Q}$ and $\underline{R}$, are chosen to satisfy the following design criteria:

1. The overshoot to a step input should be less than five percent.
2. The five percent settling time, $t_{5\%}$, is less than or equal to 1 second.

Using an iterative process, the weighting matrices that meet the above design criteria were found to be:

$$\underline{Q}=\begin{bmatrix} 0.3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} ; \qquad \underline{R}=[1] \ . \tag{4.13}$$

The resulting steady-state feedback gain matrix, $\underline{K}$, is:

$$\underline{K}=[k_1 \ k_2 \ k_3 \ k_4] = [0.5347 \ -0.2385 \ 0.1329 \ 0.0017] \ . \tag{4.14}$$

Figure 4.1 shows the roll rate controller block diagram.

d. Simulation Results

Figure 4.2(a) shows the response of the closed loop system to a step input of six degrees/second. The design criteria of an overshoot less than five percent and the five percent settling time, $t_{5\%}$, less than one second are achieved. Figure 4.2(b) shows the aileron vane angle as a result of the step input. Weir [Ref. 16] demonstrates that the control vanes *stall* when displaced by an angle of plus or minus 30 degrees from the air flow zero reference.

Figure 4.1 Full Order Roll Rate Controller Block Diagram

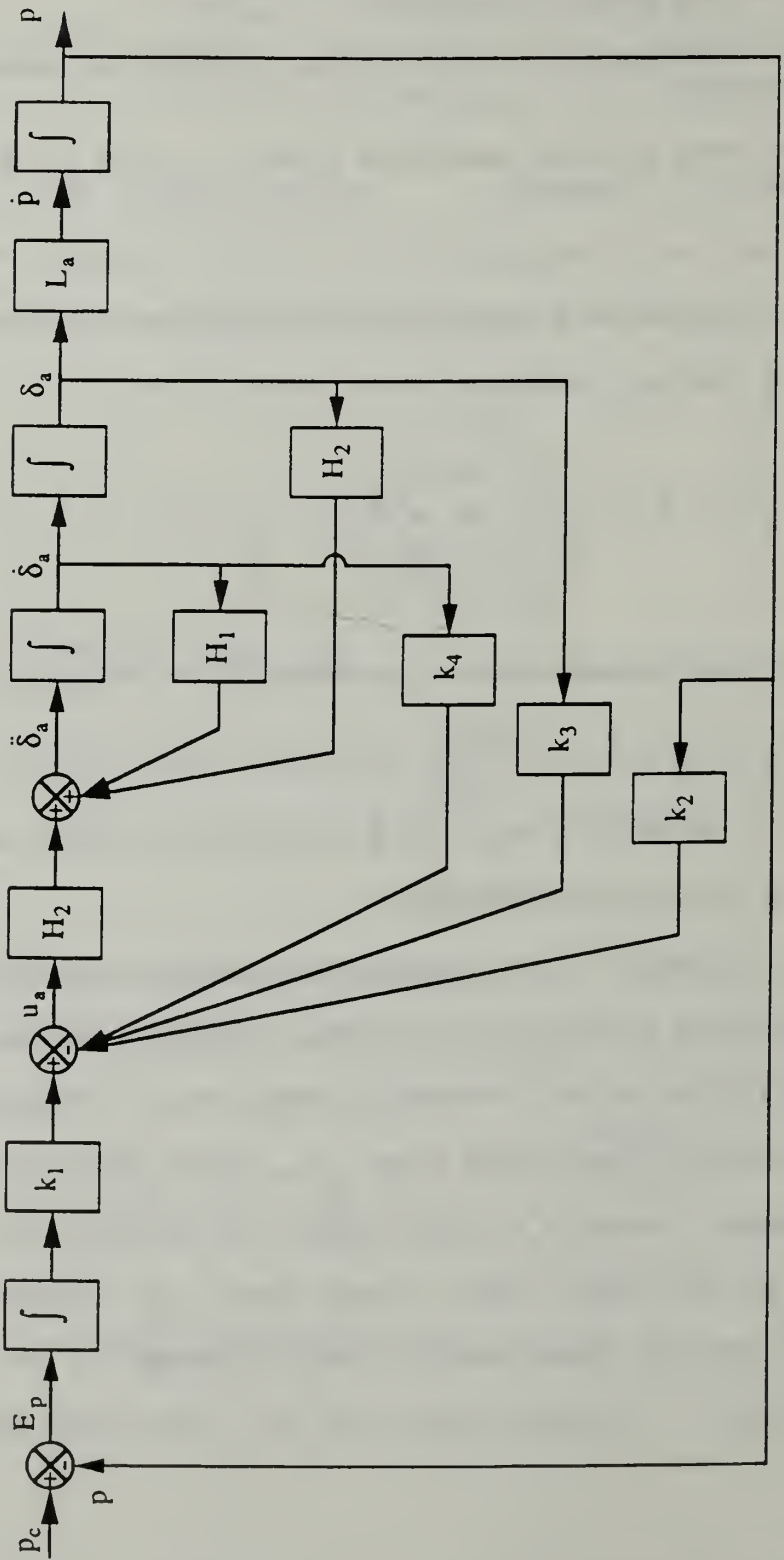
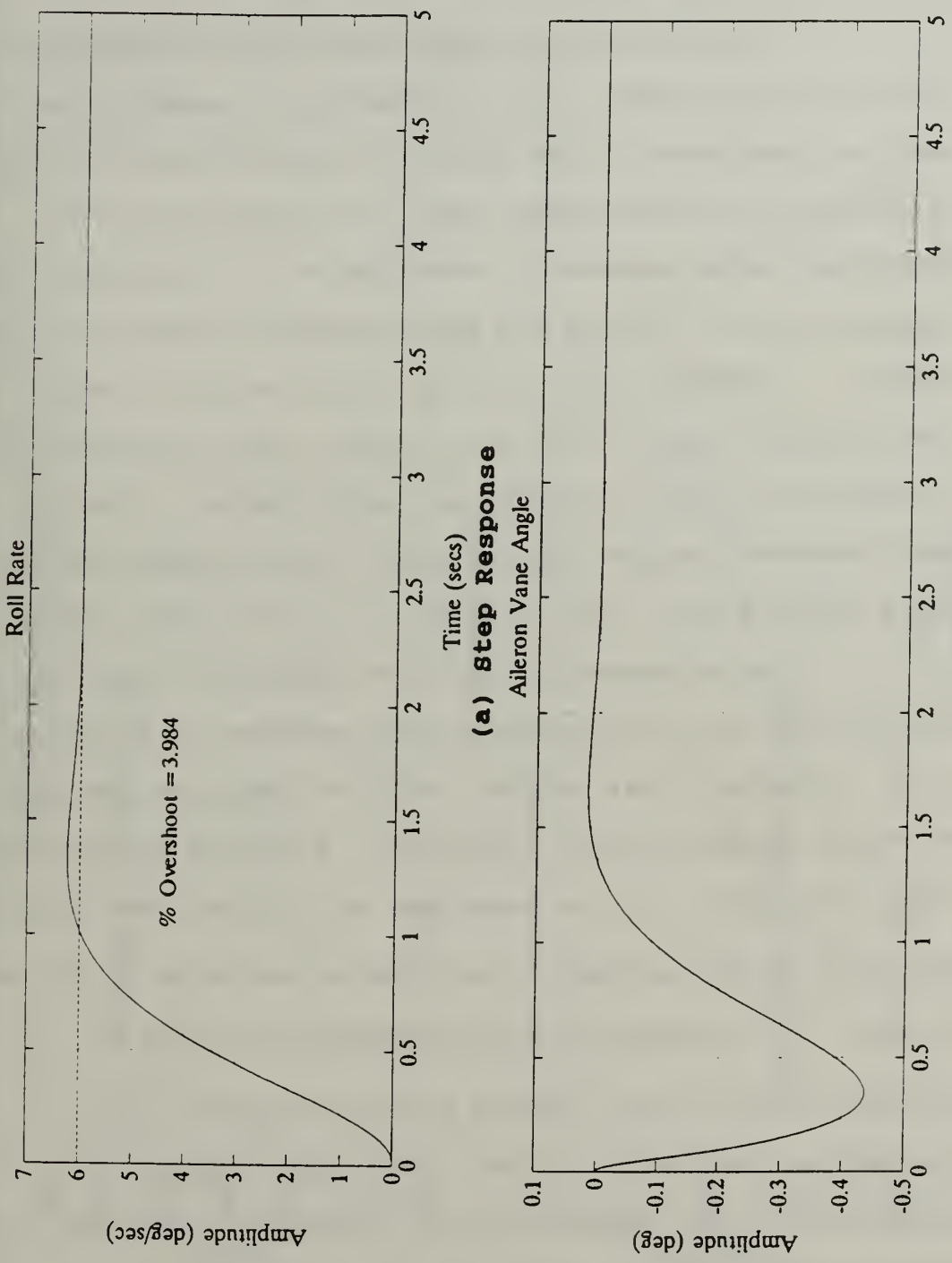


Figure 4.2 Full Order Controller Step Response

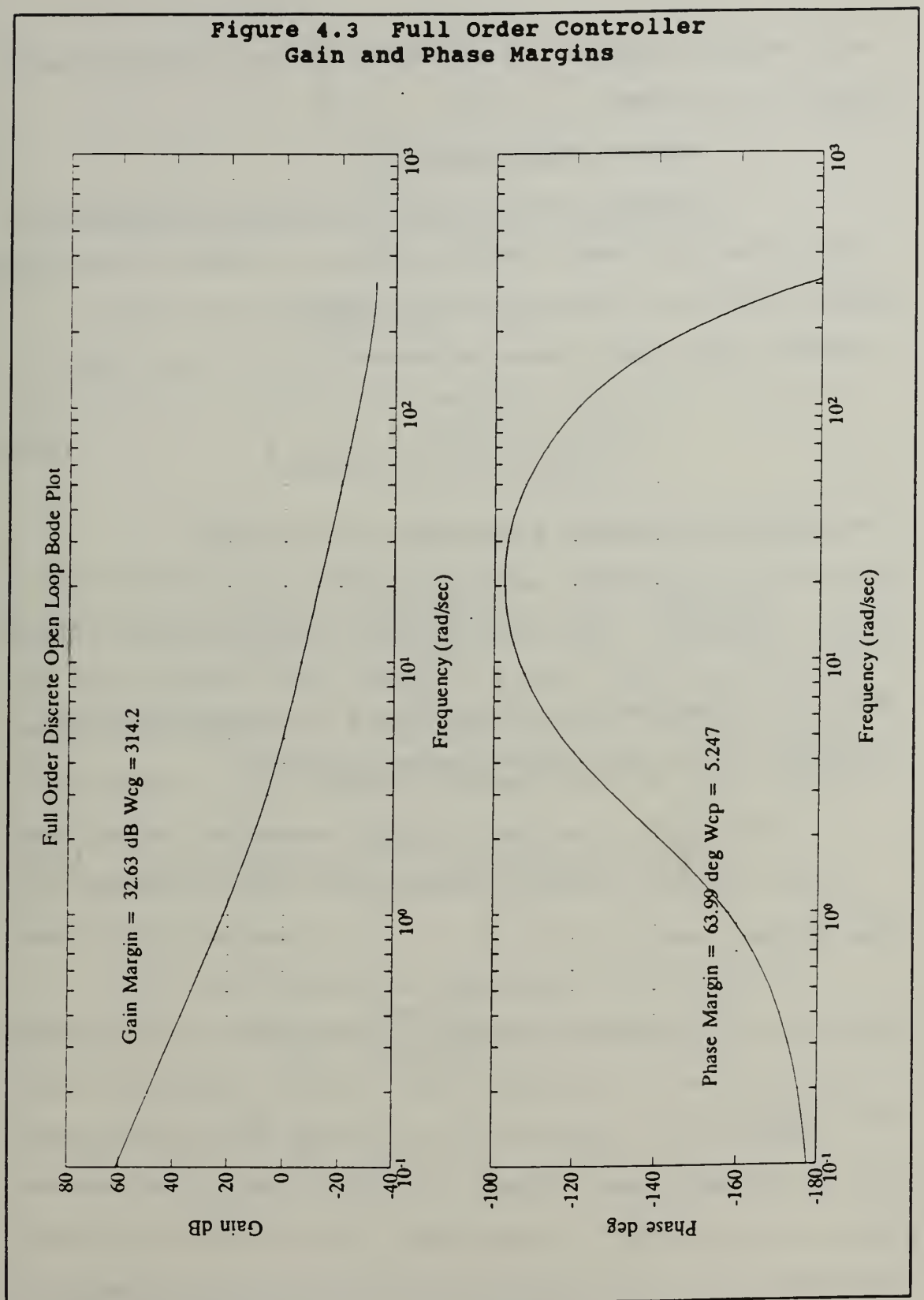


Therefore, it is important to ensure that the design not require vane angles greater than 30 degrees.

Because the roll rate controller was designed using the linear model, it is necessary to measure the relative robustness of the design to ensure that it will be applicable to the nonlinear model. The phase and gain margins are such measures. From Figure 4.3, the gain margin is equal to 32.63 dB and the phase margin is equal to 63.99 degrees. A general rule of thumb is to require a gain margin greater than six dB and a phase margin greater than 30 degrees to ensure satisfactory performance. Clearly, these benchmark values are exceeded and the above design should perform well when applied to the nonlinear model.

The steady-state gains of Equation (4.14) are an optimal solution for the performance measure of Equation (4.10). However, this optimal solution requires that all of the states be measured or estimated. Although measurement of the vane angle, δ_a, is possible, a computational observer would have to be included to provide an estimate of the vane velocity, $\dot{\delta}_a$. Inclusion of an observer would add an additional task to the onboard digital computer, and increase the complexity of the controller design. An alternative to the observer would be simply to set the vane velocity gain to zero. The system would have to be simulated with this gain equal to zero and the phase and

Figure 4.3 Full Order Controller Gain and Phase Margins



gain margins computed to determine how the system response would be affected.

e. Reduced Order Model

Because the servo dynamics are significantly faster than all other system dynamics, a second alternative is to ignore the servo dynamics completely and form a reduced order state space representation of the system.

$$\begin{bmatrix} \dot{E}_p \\ \ddot{p} \end{bmatrix} = \begin{bmatrix} 0 & -1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} E_p \\ \dot{p} \end{bmatrix} + \begin{bmatrix} 0 \\ -21.29 \end{bmatrix} \delta_a \ . \qquad (4.15)$$

The discrete-time state equation is given as

$$\begin{bmatrix} E_p(k+1) \\ \dot{p}(k+1) \end{bmatrix} = \begin{bmatrix} 1.0 & -0.01 \\ 0 & 1.0 \end{bmatrix} \begin{bmatrix} E_p(k) \\ \dot{p}(k) \end{bmatrix} + \begin{bmatrix} 0.0011 \\ -0.2129 \end{bmatrix} \delta_a(k) \ . \qquad (4.16)$$

Now, the steady-state gain matrix, $\underline{\underline{K}}$, is determined from Equation (4.16) and the performance measure:

$$J = \sum_{k=0}^{\infty} \left\{ [E_p(k) \ \dot{p}(k)] \underline{\underline{Q}} \begin{bmatrix} E_p(k) \\ \dot{p}(k) \end{bmatrix} + \dot{u}^T(k) \underline{\underline{R}} \, \dot{u}(k) \right\} \ . \qquad (4.17)$$

The optimal control $\dot{u}(k)$ is:

$$\dot{u}(k) = -[k_1 \ k_2] \begin{bmatrix} E_p(k) \\ \dot{p}(k) \end{bmatrix} = -\underline{\underline{K}} \, \underline{\underline{e}}(k) \ . \qquad (4.18)$$

and u(k) is

$$u(k) = \sum_{m=0}^{k} \dot{u}(m) = -\sum_{m=0}^{k} [k_1 \; k_2] \begin{bmatrix} E_p(m) \\ \dot{p}(m) \end{bmatrix} = -k_1 \sum_{m=0}^{k} E_p(m) - k_2 p(m) \qquad (4.19)$$

The weighting matrices, $\underline{Q}$ and $\underline{R}$, are defined as

$$\underline{Q} = \begin{bmatrix} 0.3 & 0 \\ 0 & 0 \end{bmatrix} ; \qquad \underline{R} = [1] \; . \qquad (4.20)$$

The resulting steady-state feedback gain matrix, $\underline{K}$, is

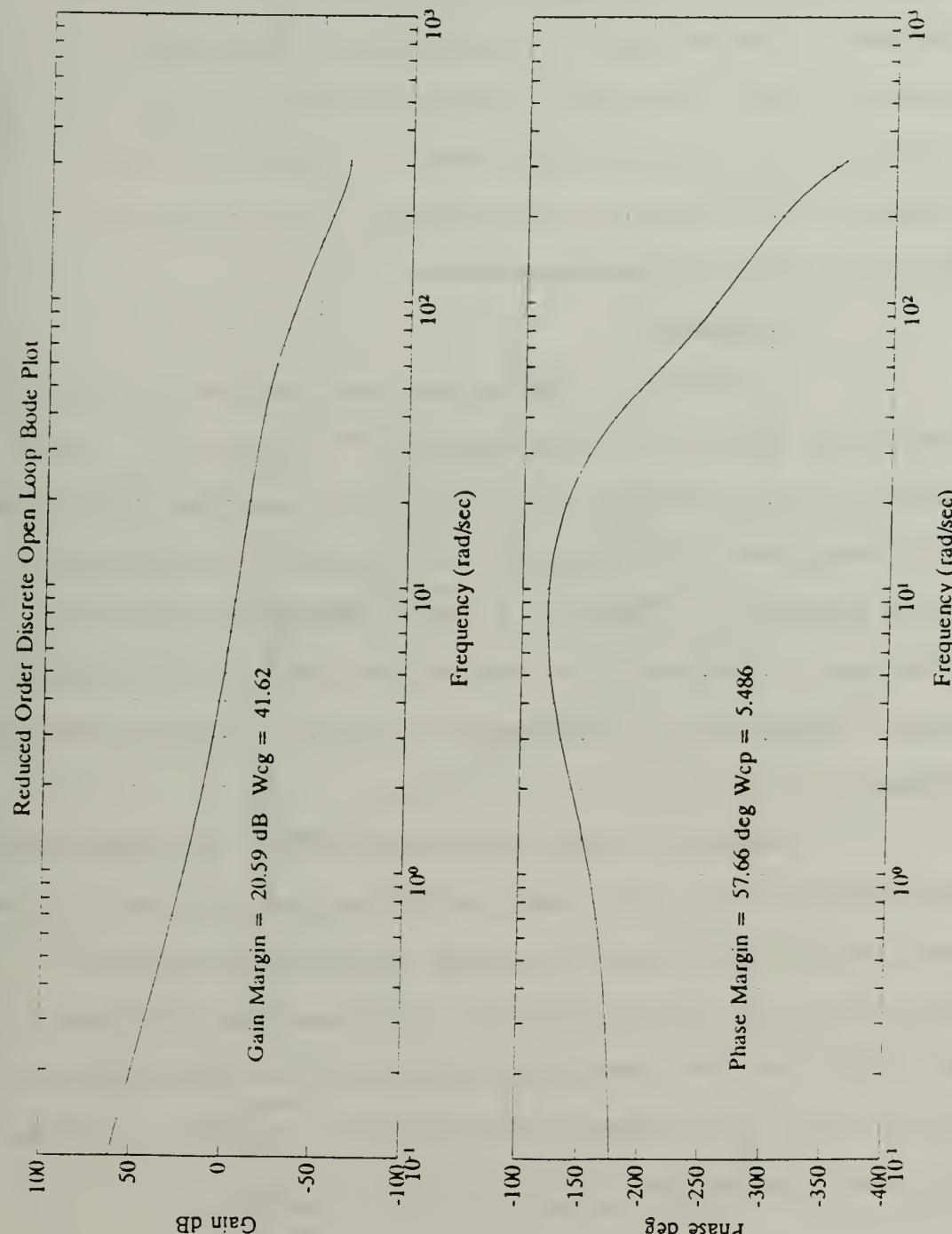
$$\underline{K} = [0.4376 \; -0.2027] \; . \qquad (4.21)$$

Figure 4.4(a) shows the response of the system to a step input of six degrees/second. The design criteria for overshoot and settling time are achieved. The step response for this reduced order model is almost identical to the response for the full order model given in Figure 4.2(a). From Figure 4.4(b), the maximum magnitude of the aileron vane angle displacement is 0.47 degrees. This vane displacement angle meets the requirement of less than or equal to 30 degrees.

The necessity of determining the relative robustness of the system is now magnified due to the reduced order controller design. From Figure 4.5, the gain and phase margins are equal to 20.59 dB and 57.66 degrees respectively. Note that these margins were computed for the controller with the full order model. The reduction in gain and phase margins from the full order model is an expected

Figure 4.4 Reduced Order Controller Step Response

Figure 4.5 Reduced Order Controller Phase and Gain Margins



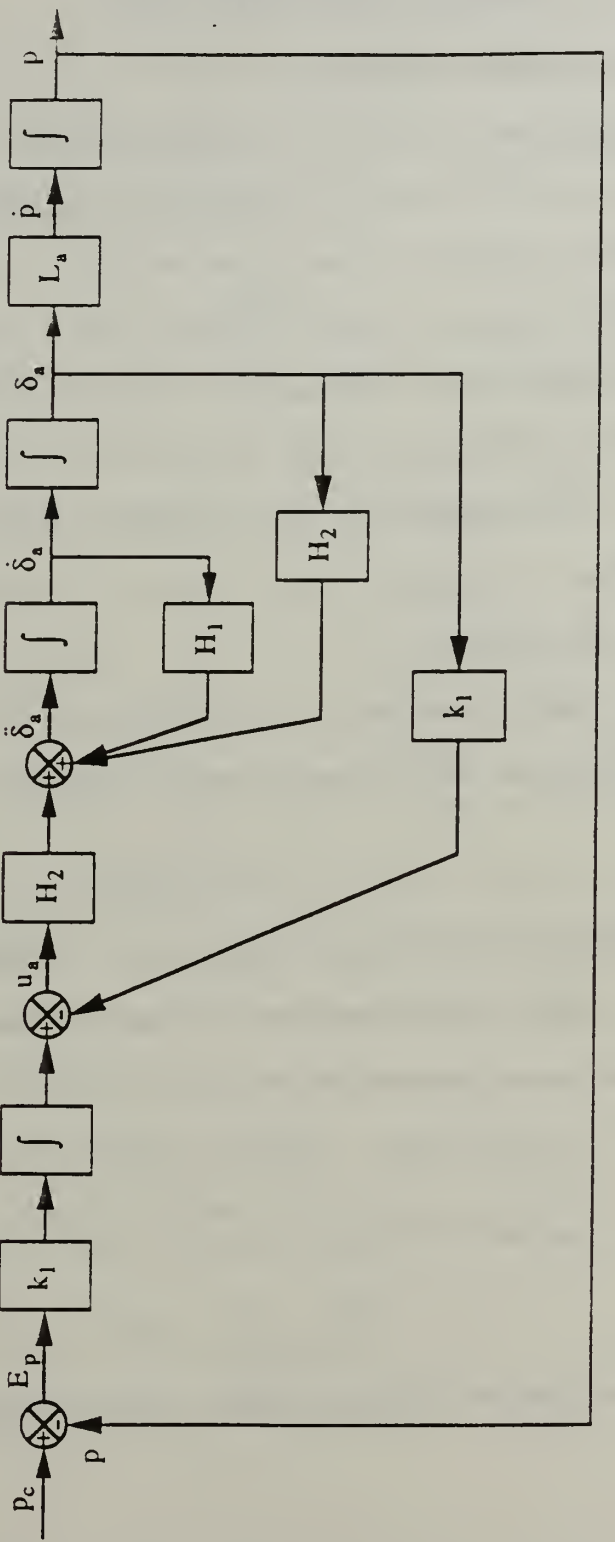
consequence of the reduced order controller. However, the requirement for a minimum of six dB gain margin and 30 degrees of phase margin is satisfied. Therefore, it is expected that the reduced order controller design will be applicable to the nonlinear model. Figure 4.6 is a graphical realization of the reduced order controller applied to the roll rate system.

f. Summary

An optimal tracker was designed for the roll rate control by applying the procedures of Chapter III. This control system yielded very satisfactory system performance and robustness. Recognizing the dynamics of the servos could possibly be ignored, a second reduced order design was developed. The resulting controller proved to be robust and almost identical in performance to the full order controller design.

Because of the favorable results obtained with the reduced order roll rate controller design, the altitude rate and the pitch and yaw angle controllers will be designed using this technique. The resulting altitude rate and pitch and yaw controllers will be evaluated analogously to the roll rate controller to ensure they meet or exceed the design criteria.

Figure 4.6 Reduced Order Roll Rate Controller Block Diagram



C. ARCHYTAS ALTITUDE RATE CONTROLLER

1. The Altitude System

The primary function of the altitude rate controller for Archytas is to allow the operator to command a desired translational velocity along the vehicle's longitudinal, or x axis. This translational velocity will permit the operator to position Archytas at a given altitude by terminating the velocity when a desired altitude is achieved. This capability is critical during the landing phase in order to control the altitude as Archytas lands in a vertical position.

The simplified equation of motion which describes Archytas' altitude rate subsystem is given as:

$$\dot{u} = \ddot{h} = 0.10029\ \delta_{rpm} \tag{4.22}$$

Note the change of notation from $\dot{u}$ to $\ddot{h}$, which is more appropriate. The throttle servo model is identical to the aileron servo model and is given as:

$$\dot{\delta}_t = -74.1\delta_t - 2745.8\delta_t + 2745.8u_t\ . \tag{4.23}$$

The Archytas engine is modeled as a first order lag:

$$\dot{\delta}_{rpm} = -2\delta_{rpm} + 1675.52\delta_t\ . \tag{4.24}$$

The altitude rate tracking error is defined as:

$$E_{\dot{h}} = \dot{h}_c - \dot{h} \ ; \tag{4.25}$$

where $\dot{h}_c$ is the input command and $\dot{h}$ is the measured altitude rate. From Equation (4.25), the differential equation for the tracking error is:

$$\dot{E}_{\dot{h}} = -\ddot{h} \ . \tag{4.26}$$

Combining Equation (4.26) with the time derivatives of Equations (4.22), (4.23) and (4.24), a state equation that is appropriate for tracking system design is:

$$\begin{bmatrix} \dot{E}_{\dot{h}} \\ \ddot{h} \\ \dot{\delta}_{rpm} \\ \dot{\delta}_t \\ \ddot{\delta}_t \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & -0.10029 & 0 & 0 \\ 0 & 0 & -2 & 1675.5 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -2745.8 & -74.1 \end{bmatrix} \begin{bmatrix} E_{\dot{h}} \\ \ddot{h} \\ \dot{\delta}_{rpm} \\ \dot{\delta}_t \\ \ddot{\delta}_t \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 2745.8 \end{bmatrix} \dot{u}_t \ . \tag{4.27}$$

In a manner identical to that used for the roll rate controller design, the reduced order altitude rate model is formed. The controller is designed using the reduced order model and its performance and robustness are evaluated with respect to the full order model. Ignoring the servo dynamics, the reduced order state space model is:

$$\begin{bmatrix} \dot{E}_{\dot{h}} \\ \ddot{h} \\ \dot{\delta}_{rpm} \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & 0.10029 \\ 0 & 0 & -2 \end{bmatrix} \begin{bmatrix} E_{\dot{h}} \\ \ddot{h} \\ \dot{\delta}_{rpm} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1675.5 \end{bmatrix} \dot{u}_t \ . \tag{4.28}$$

2. Altitude Rate Controller Design

a. Discretizing the Altitude Rate System

With the sampling period, T, equal to 0.01 seconds, the discrete-time state equation is:

$$\begin{bmatrix} E_{\dot{h}}(k+1) \\ \ddot{h}(k+1) \\ \delta_{rpm}(k+1) \end{bmatrix} = \begin{bmatrix} 1 & -0.01 & 0 \\ 0 & 1 & 0.001 \\ 0 & 0 & 0.9802 \end{bmatrix} \begin{bmatrix} E_{\dot{h}}(k) \\ \ddot{h}(k) \\ \delta_{rpm}(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1675.5 \end{bmatrix} \dot{u}_c \quad . \tag{4.29}$$

b. Gain Determination

The LQR weighting matrices, $\underline{Q}$ and $\underline{R}$, for the altitude rate control loop are chosen to satisfy the following design criteria:

1. The overshoot to a step input should be less than five percent.
2. The five percent settling time, $t_{5\%}$, is less than or equal to 2 seconds.

The resulting weighting matrices are:

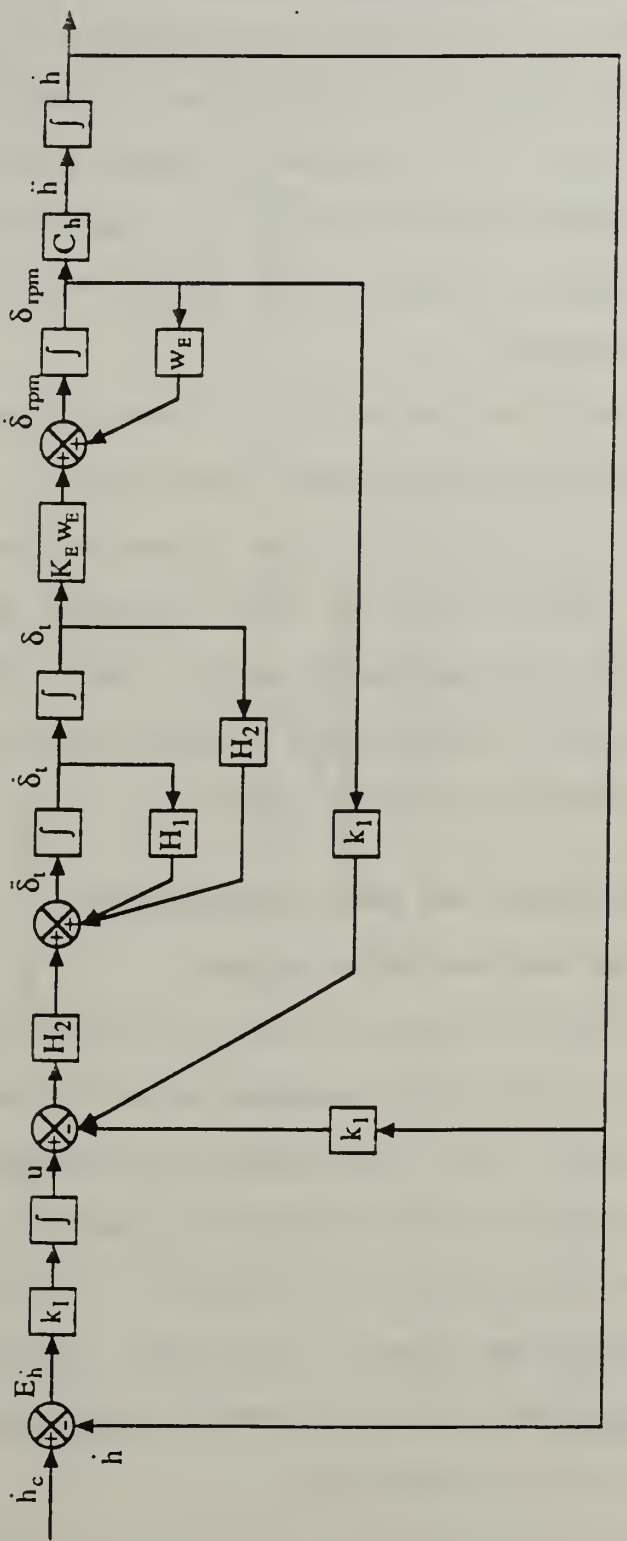
$$\underline{Q} = \begin{bmatrix} 0.1 & 0 & 0 \\ 0 & 0.01 & 0 \\ 0 & 0 & 0 \end{bmatrix} ; \qquad \underline{R} = [1] \quad . \tag{4.30}$$

The steady-state feedback gain matrix, $\underline{K}$, is:

$$\underline{K} = [-0.3060 \;\; 0.2003 \;\; 0.0038] \quad . \tag{4.31}$$

Figure 4.7 shows the altitude rate controller block diagram.

Figure 4.7 Reduced Order Altitude Rate Block Diagram



c. Simulation Results

Figure 4.8(a) shows the response of the altitude rate closed loop system to a step input of ten feet/second. The design criteria of an overshoot less than five percent and the five percent settling time, $t_{5\%}$, less than two seconds is achieved. Figure 4.8(b) shows the control applied to the system.

The gain and phase margins were determined as a measure of the system robustness. From Figure 4.9, the gain margin is equal to 18.03 dB and the phase margin is equal to 58.78 degrees. These values of gain and phase margin indicate that the altitude rate control system design based on the reduced order model should provide good results when applied to the nonlinear system model.

D. ARCHYTAS PITCH AND YAW ANGLE CONTROLLER

1. The Pitch and Yaw Angle System

The purpose of the pitch and yaw angle controller for Archytas is to maintain commanded orientation around the pitch and yaw axes. This requirement is necessary to allow the operator to position Archytas during landing by pitching or yawing the vehicle slightly to induce a translational velocity along the body fixed y or z axis. This control problem is complicated by the gyroscopic coupling between the axes caused by the propeller.

Figure 4.8 Reduced Order Altitude Rate Controller Step Response

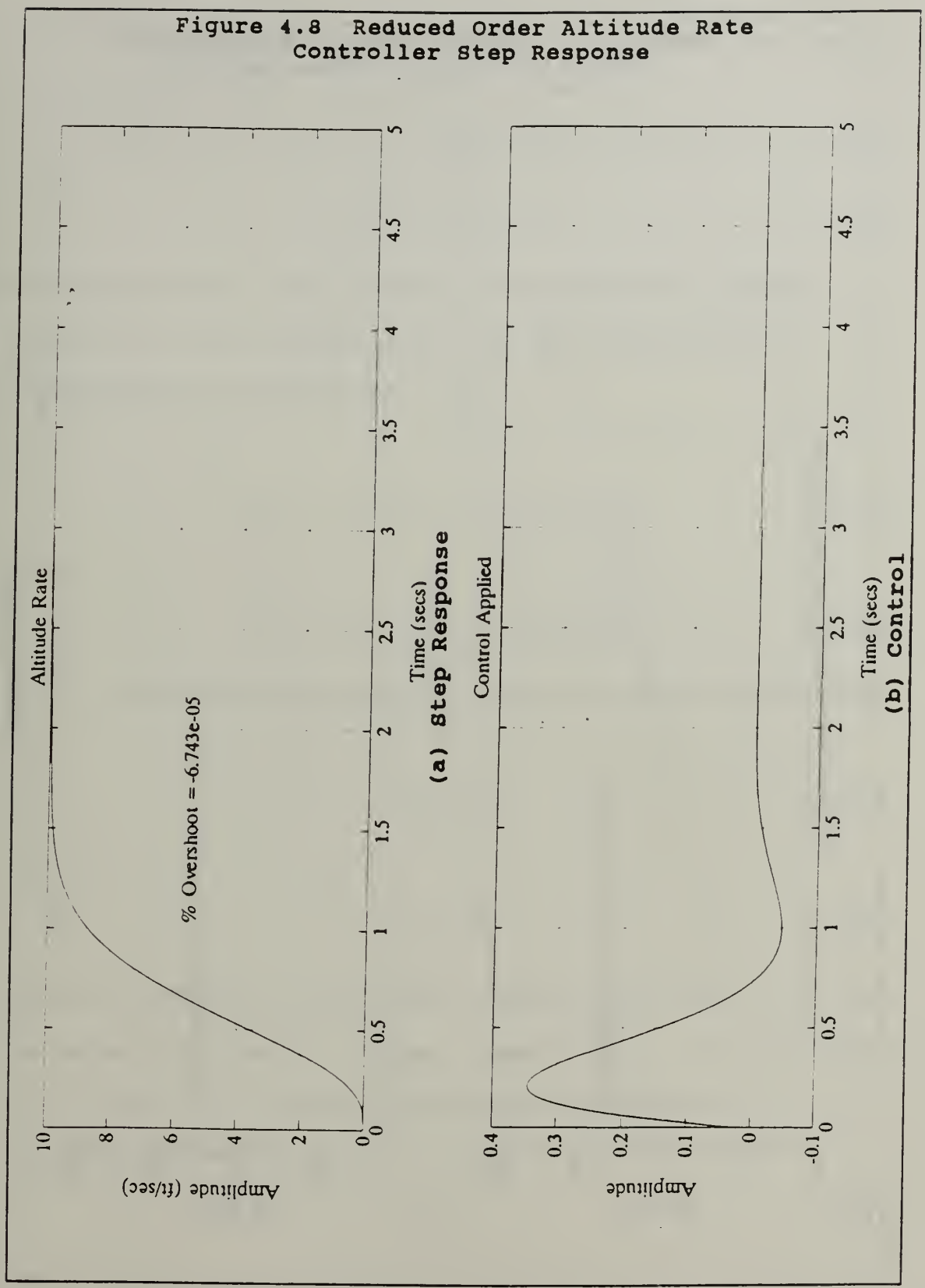
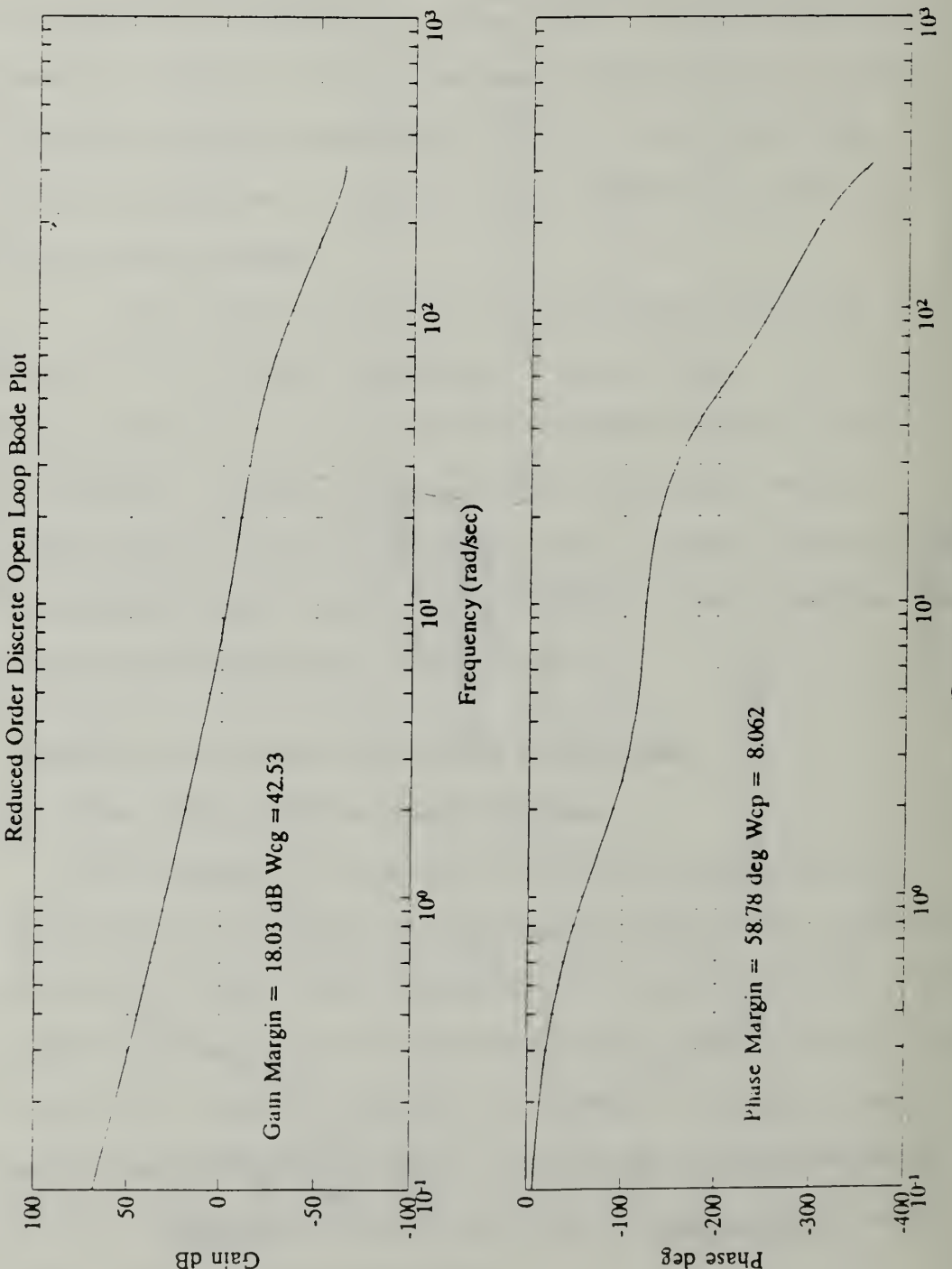


Figure 4.9 Reduced Order Altitude Rate Controller Gain and Phase Margins



The simplified equations of motion that describe the pitch and yaw angle subsystem are given as:

$$\dot{q}=-11.19\delta_e-1.62r; \tag{4.32}$$

$$\dot{r}=-12.28\delta_r+1.78q\ . \tag{4.33}$$

The elevator and rudder servos are modeled in a manner identical to the aileron servo and the corresponding differential equations are:

$$\ddot{\delta}_e=-74.1\dot{\delta}_e-2745.8\delta_e+2745.8u_e\ ; \tag{4.34}$$

and

$$\ddot{\delta}_r=-74.1\dot{\delta}_r-2745.8\delta_r+2745.8u_r\ . \tag{4.35}$$

The pitch and yaw angle tracking errors are defined as:

$$E_\theta=\theta_c-\theta\ ; \tag{4.36}$$

and

$$E_\Psi=\Psi_c-\Psi\ ; \tag{4.37}$$

where θ_c and Ψ_c are the input commands and θ and Ψ are the measured pitch and yaw angles, respectively. From Equations (4.36) and (4.37), the differential equations for the tracking errors are:

$$\dot{E}_\theta=-q\ ; \tag{4.38}$$

and

$$\dot{E}_{\Psi} = -r \quad . \tag{4.39}$$

Combining Equations (4.38) and (4.39) with the time derivatives of Equations (4.34) and (4.35), a state equation that is appropriate for tracking system design is $\dot{\underline{x}} = \underline{\underline{A}}\,\underline{x} + \underline{\underline{B}}\,\underline{v}$, where the state is:

$$\underline{x}^T = [E_\theta \; \dot{E}_\theta \; \dot{q} \; E_\Psi \; \dot{E}_\Psi \; \dot{r} \; \delta_e \; \delta_{\dot{e}} \; \delta_{\dot{e}} \; \delta_r \;]^T \tag{4.40}$$

the control matrix is $\underline{u}^T = [u_e \; u_{\dot{e}}]^T$, and the system matrices are:

$$\underline{\underline{A}} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1.62 & -11.19 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.78 & 0 & 0 & 0 & 0 & -12.28 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -2745.8 & 0 & -74.1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2745.8 & 0 & -74.1 \end{bmatrix} \tag{4.41}$$

and

$$\underline{\underline{B}}^T = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2745.8 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2745.8 \end{bmatrix}^T \quad . \tag{4.42}$$

In a manner similar to that used in the roll rate and altitude rate controller designs, the reduced order model is formed by ignoring the servo dynamics. The resulting state is:

$$\underline{x}^T = [E_\theta \ \dot{E}_\theta \ \dot{q} \ E_\Psi \ \dot{E}_\Psi \ \dot{r}]^T \ ; \tag{4.43}$$

the control matrix is $\underline{v}^T = [\delta_e \ \delta_r]^T$, and the system matrices are:

$$\underline{\underline{A}} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1.62 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1.78 & 0 & 0 & 0 \end{bmatrix} ; \tag{4.44}$$

and

$$\underline{\underline{B}} = \begin{bmatrix} 0 & 0 & -11.19 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -12.28 \end{bmatrix}^T . \tag{4.45}$$

2. Pitch and Yaw Angle Controller Design

a. Discretizing the Pitch and Yaw Angle System

Using the sampling period, T, equal to 0.01 seconds, the discrete-time state is:

$$\underline{x}(k)^T = [E_\theta(k) \ \dot{E}_\theta(k) \ \dot{q}(k) \ E_\Psi(k) \ \dot{E}_\Psi(k) \ \dot{r}(k)]^T \ ; \tag{4.46}$$

the control matrix is $\underline{v}(k)^T = [\delta_e(k) \ \delta_r(k)]^T$, and the discrete-time system matrices are:

$$\underline{\underline{\Phi}}(k) = \begin{bmatrix} 1 & 0.01 & 0 & 0 & 0 & 0 \\ 0 & 1 & -0.01 & 0 & 0 & 0.0001 \\ 0 & 0 & 0.9999 & 0 & 0 & -0.0162 \\ 0 & 0 & 0 & 1.0101 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.99 & 0 \\ 0 & 0 & 0.0178 & 0 & 0 & 0.9999 \end{bmatrix} \quad \textbf{(4.47)}$$

and

$$\Gamma(k) = \begin{bmatrix} 0 & 0 \\ 0.0006 & 0 \\ -0.1119 & 0.001 \\ 0 & 0 \\ 0 & 0 \\ -0.001 & -0.1228 \end{bmatrix}. \quad \textbf{(4.48)}$$

b. Gain Determination

The LQR weighting matrices, $\underline{\underline{Q}}$ and $\underline{\underline{R}}$, for the pitch and yaw angle control loop are chosen to satisfy the following design criteria:

1. The overshoot to a step response should be less than 50 percent.
2. The five percent settling time, $t_{5\%}$, is less than or equal to 2 seconds.

The allowable overshoot requirement may seem too liberal. However, due to the small pitch or yaw angles (one to five degrees) required to induce translational velocities along the body fixed y and z axes, the overshoot should not present any problems to the operator. Computer simulations have indicated that excessive control values, and vane angle

deflections would be required, if the controller was designed to limit the overshoot to values less than 15 percent. The large overshoot values are due to the reduced order controller. If a full order controller is implemented, the large overshoot values could be eliminated with small vane angles and a reasonable amount of control. However, that would require additional sensors and/or an observer as discussed previously.

The resulting state weighting matrices were found to be:

$$\underline{\underline{Q}} = \begin{bmatrix} 6 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & .05 & 0 & 0 & 0 \\ 0 & 0 & 0 & 6 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & .05 \end{bmatrix}; \quad \underline{\underline{R}} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{4.49}$$

The steady-state feedback gain matrix, $\underline{\underline{K}}$, is:

$$\underline{\underline{K}} = \begin{bmatrix} 2.2741 & 1.8933 & -0.6059 & 0.6390 & 0.5320 & 0.0024 \\ -0.6390 & -0.5320 & -0.0024 & 2.2741 & 1.8933 & -0.6059 \end{bmatrix}. \tag{4.50}$$

Figure 4.10 shows the MIMO pitch angle and yaw angle controller block diagram [Ref. 14: p. 179].

c. Simulation Results

Figure 4.11(a) show the response of the system to a commanded pitch angle of ten degrees and a commanded yaw angle of zero. The large value of overshoot was expected due to the severe coupling of the axes. Figure 4.11(b)

Figure 4.10 Reduced Order Pitch Angle and Yaw Angle Controller Block Diagram

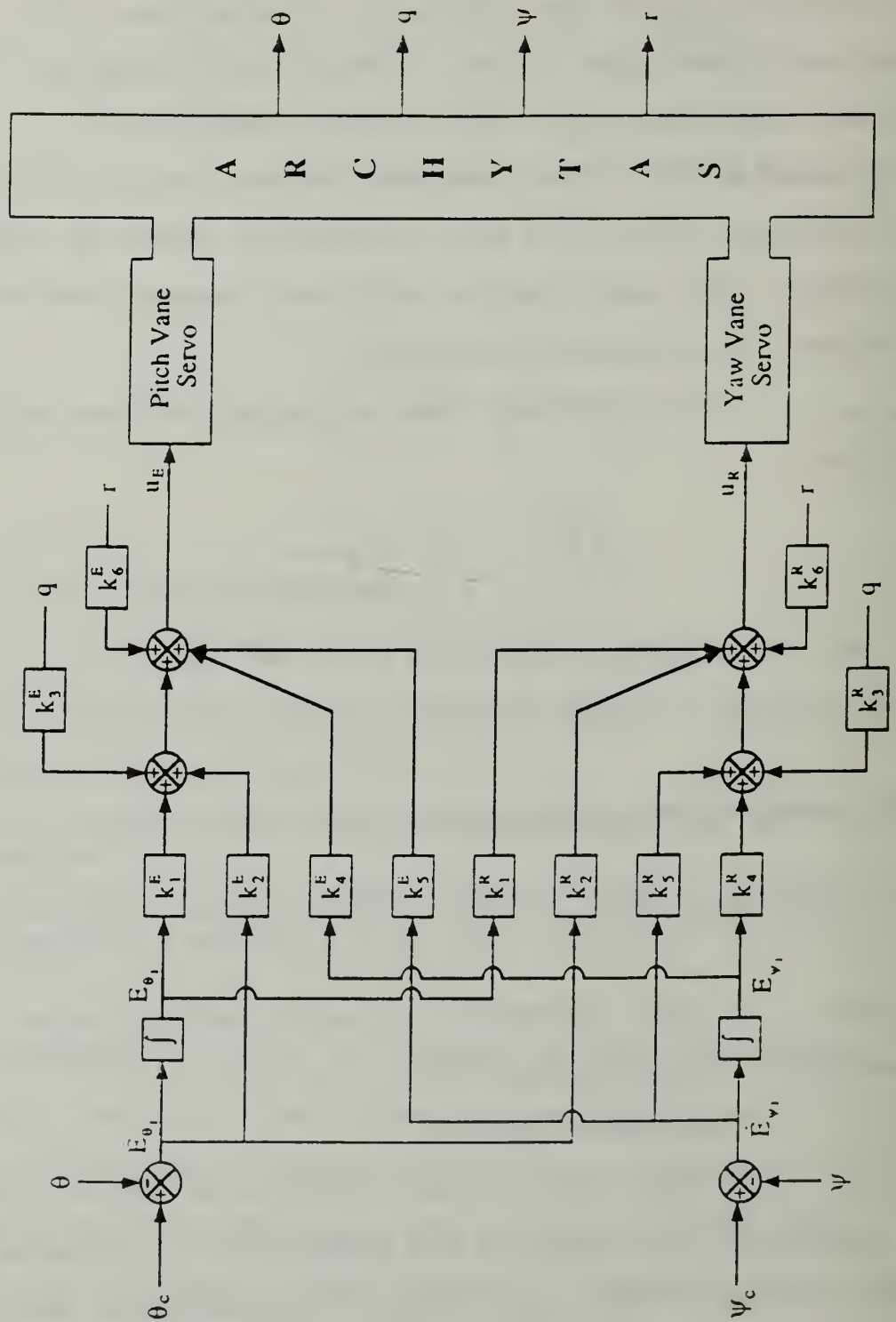
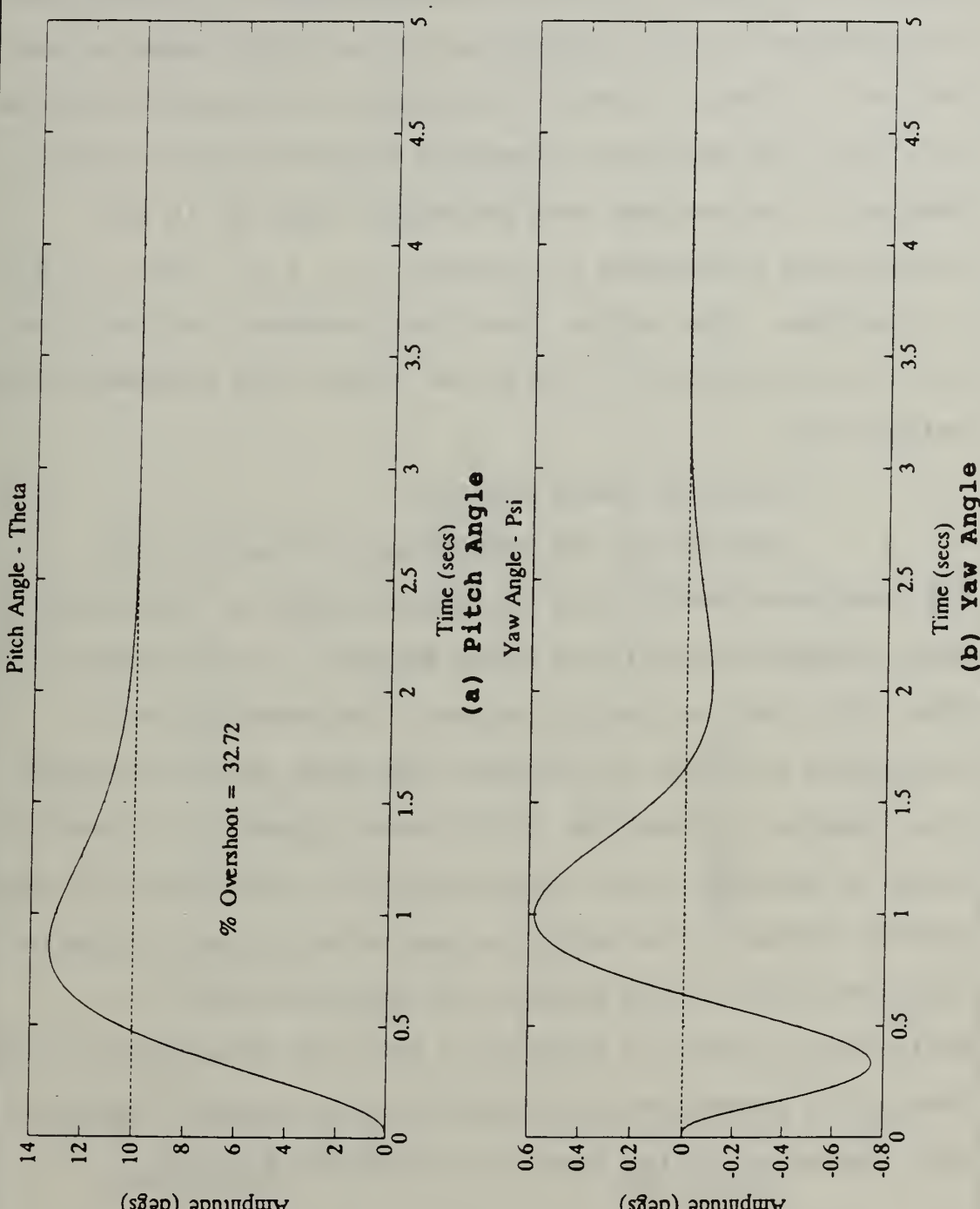


Figure 4.11 Pitch Angle Equal to Ten Degrees / Yaw Angle Equal to Zero



shows the induced yaw angle from the commanded pitch angle. Figure 4.12 gives a similar representation. The yaw angle is commanded to five degrees while the pitch angle is held at zero. Finally, Figure 4.13 shows the response when the pitch and yaw angle are commanded simultaneously to five degrees. The maximum vane deflection applied in the simulations documented by Figures 4.11, 4.12, and 4.13 was 1.7 degrees. The design goals for overshoot and settling time are satisfied for the linear model with reasonable vane deflections.

d. Singular Value Analysis

Evaluating the robustness of the roll and altitude rate controllers was accomplished by computing their respective gain and phase margins. In the case of the MIMO pitch and yaw control system, the description of robustness in terms of the gain and phase margins becomes more complex. Therefore, a different approach is taken in order to determine the robustness of the pitch and yaw angle control system. The method of choice is to apply singular value analysis to the system with perturbations. Maciejowski [Ref. 19] presents a detailed development of the theory and procedures for computing the *singular values* of a MIMO system and using them for robustness analysis.

Figure 4.12 Pitch Angle Equal to Zero / Yaw Angle Equal to Five Degrees

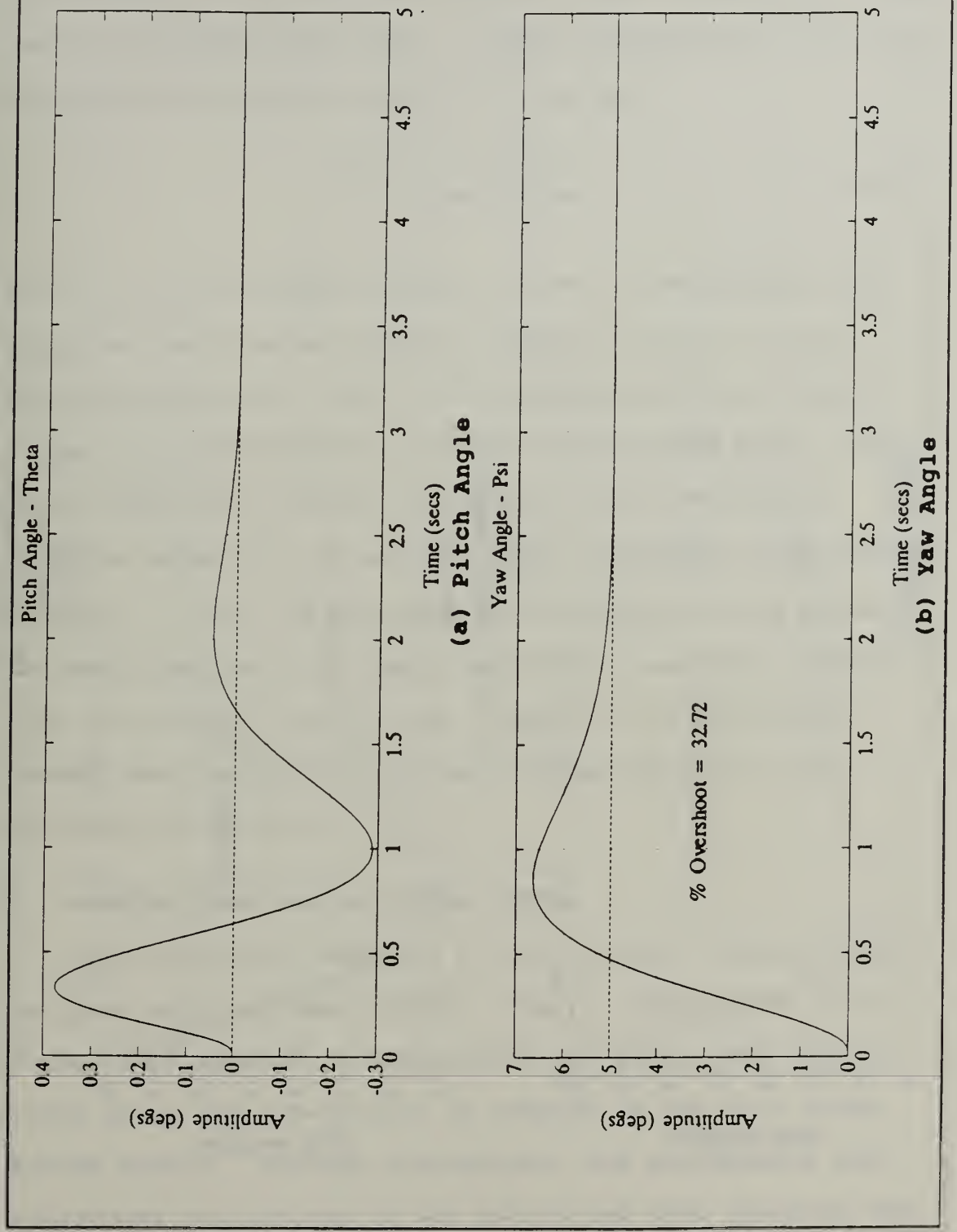


Figure 4.13 Pitch Angle Equal to Five Degrees / Yaw Angle Equal to Five Degrees

Singular value analysis is used to examine the robustness of the MIMO pitch and yaw angle controller. Burl [Ref. 13] demonstrates that a system is stable for all input multiplicative perturbations, Δ, such that

$$\|\Delta\|_\infty \leq \frac{1}{\underset{\omega}{Max}\ \bar{\sigma}\ \underline{\underline{T}}(j\omega)} \qquad (4.51)$$

where $\bar{\sigma}$ is the largest singular value of the matrix, and $\underline{\underline{T}}(j\omega)$ is the transfer function from the output of the perturbation to the input to the perturbation as shown in Figure 4.14. The singular values for the MIMO pitch and yaw angle system are plotted in Figure 4.15. The largest singular value, $\bar{\sigma}$, is equal to 1.5. Therefore, applying Equation (4.51), the MIMO system is expected to be robust for perturbations such that the infinity norm of Δ is less than or equal to 0.667. This indicates the controller design should provide good results when applied to the nonlinear model.[Ref. 13]

E. RESULTS WITH THE NONLINEAR SYSTEM

The controllers designed in the previous sections were designed using optimal control theory. Because the servo states were ignored in determining the gains, the steady-state gains are sub-optimal in respect to the full order system models. Through simulations, and performance and robustness evaluations it was determined that ignoring the

Figure 4.14 MIMO Block Diagram with Perturbations

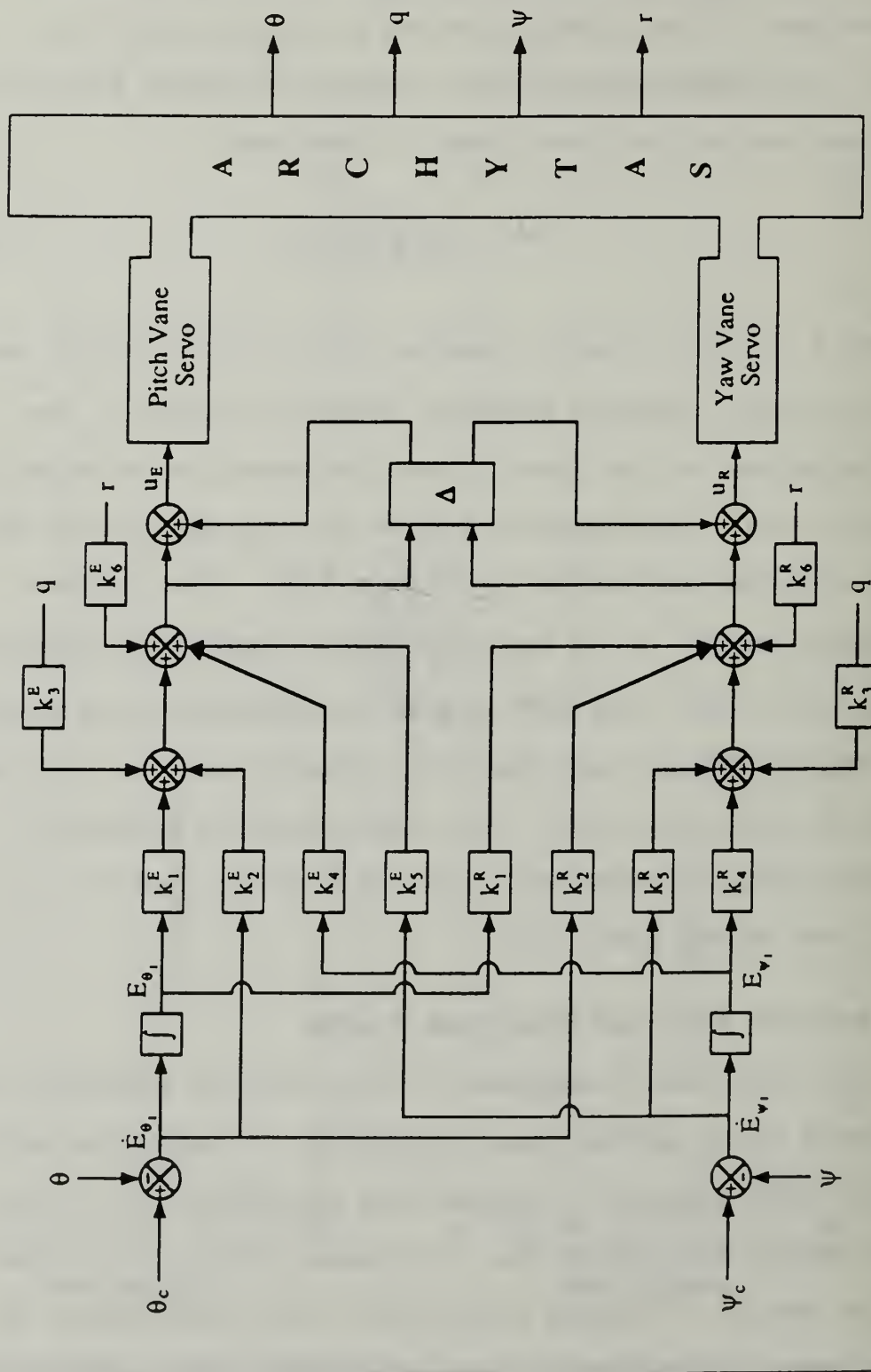
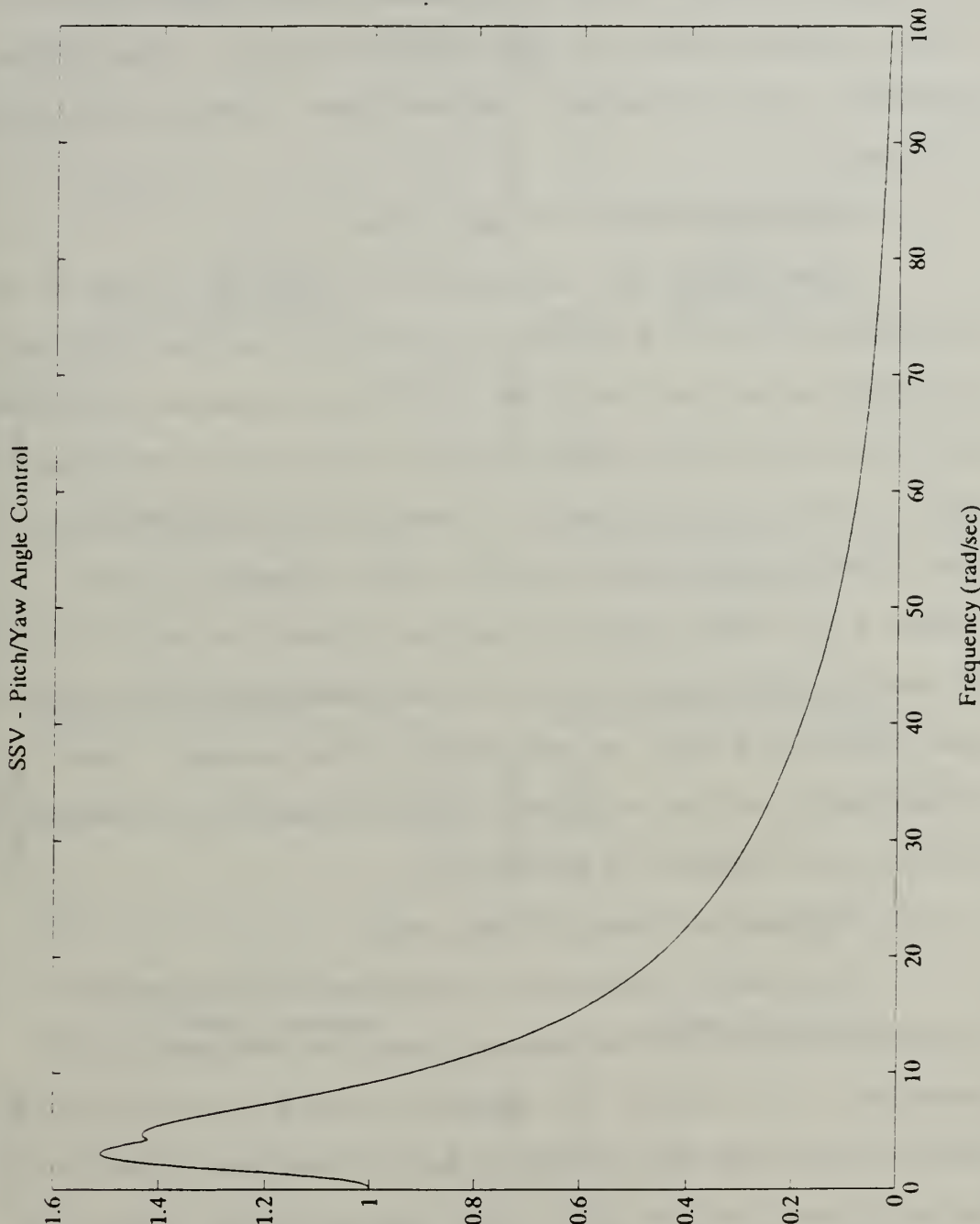


Figure 4.15 MIMO Pitch Angle and Yaw Angle Singular Values



servo dynamics would have no adverse affects. This section will apply the controllers designed with respect to the linear system models to the nonlinear model. The MATLAB programs used to simulate the nonlinear system are included as Appendix A.

1. Simulation One - Figure 4.16

The system was commanded to climb at a rate of ten feet/second for five seconds. The slope of the altitude plot indicates that the rate of ten feet/second is achieved. The roll rate, pitch angle and yaw angle were set equal to zero. The control system is tracking the commanded altitude rate input while regulating the other inputs to zero. Figure 4.16 shows that the vehicle climbs to an altitude of 50 feet. Note the coupling between the roll axis (both roll rate and roll angle) as the engine accelerates. The displacement of the aileron control vanes do not exceed plus or minus one degree in deflection.

2. Simulation Two - Figure 4.17

Figure 4.17 shows the response to a commanded altitude rate of 25 feet/second for four seconds. A commanded roll rate of ten degrees/second for two seconds is applied starting at time equal to six seconds. The slope of the altitude plot indicates that the altitude rate of 25 feet/second is achieved. The vehicle overshoots the desired 100 feet altitude and begins to approach a steady-state

Figure 4.16 Nonlinear Simulation One

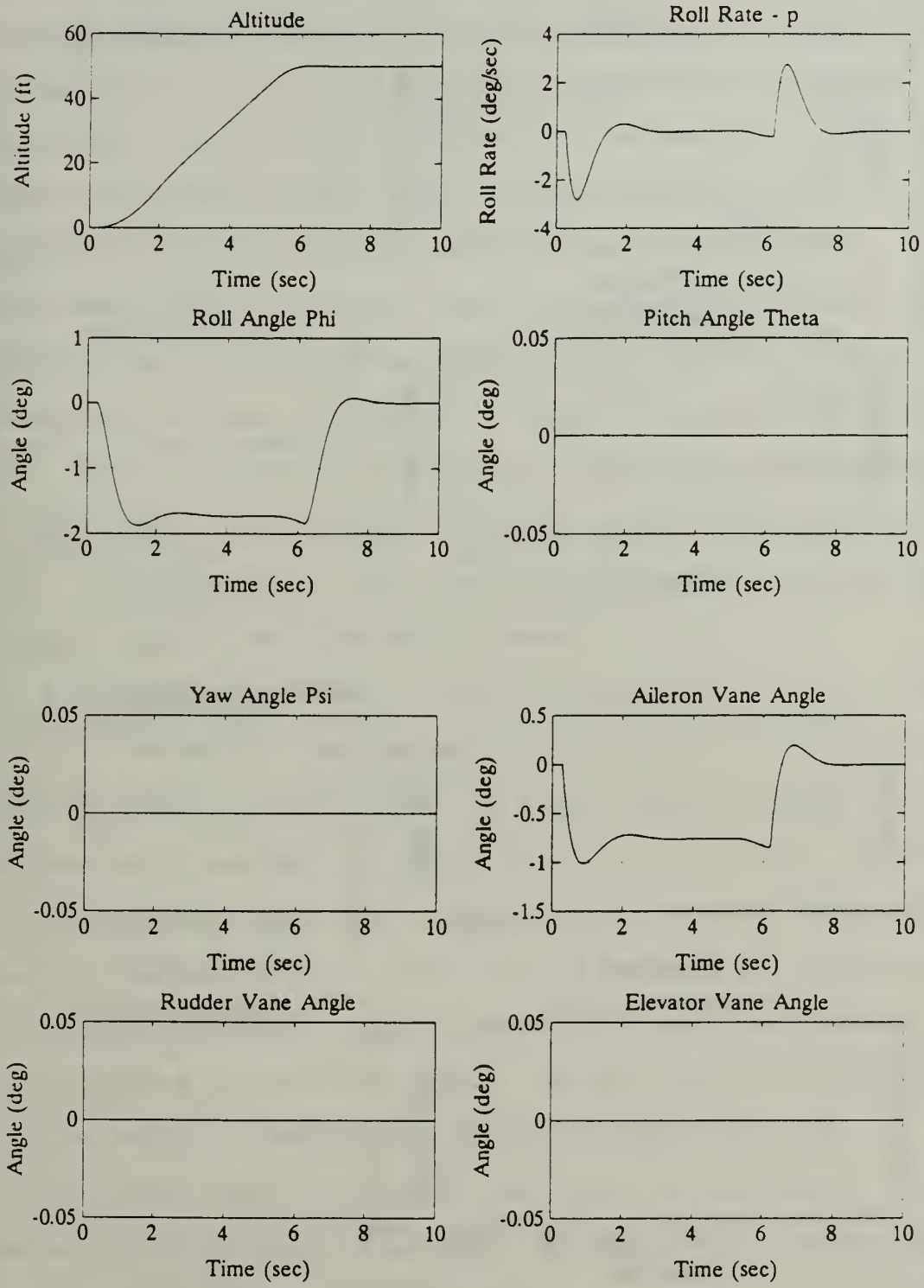
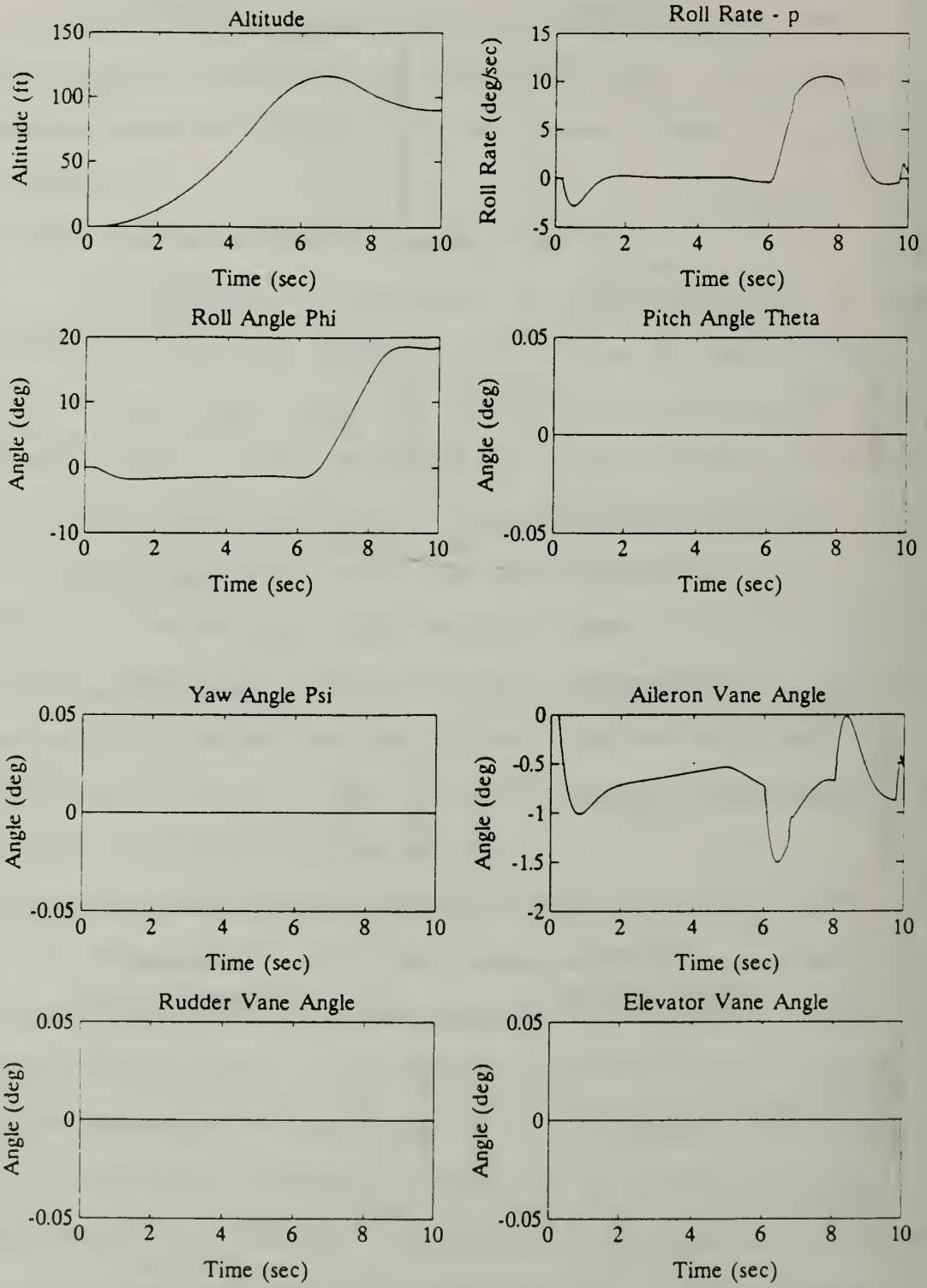


Figure 4.17 Nonlinear Simulation Two

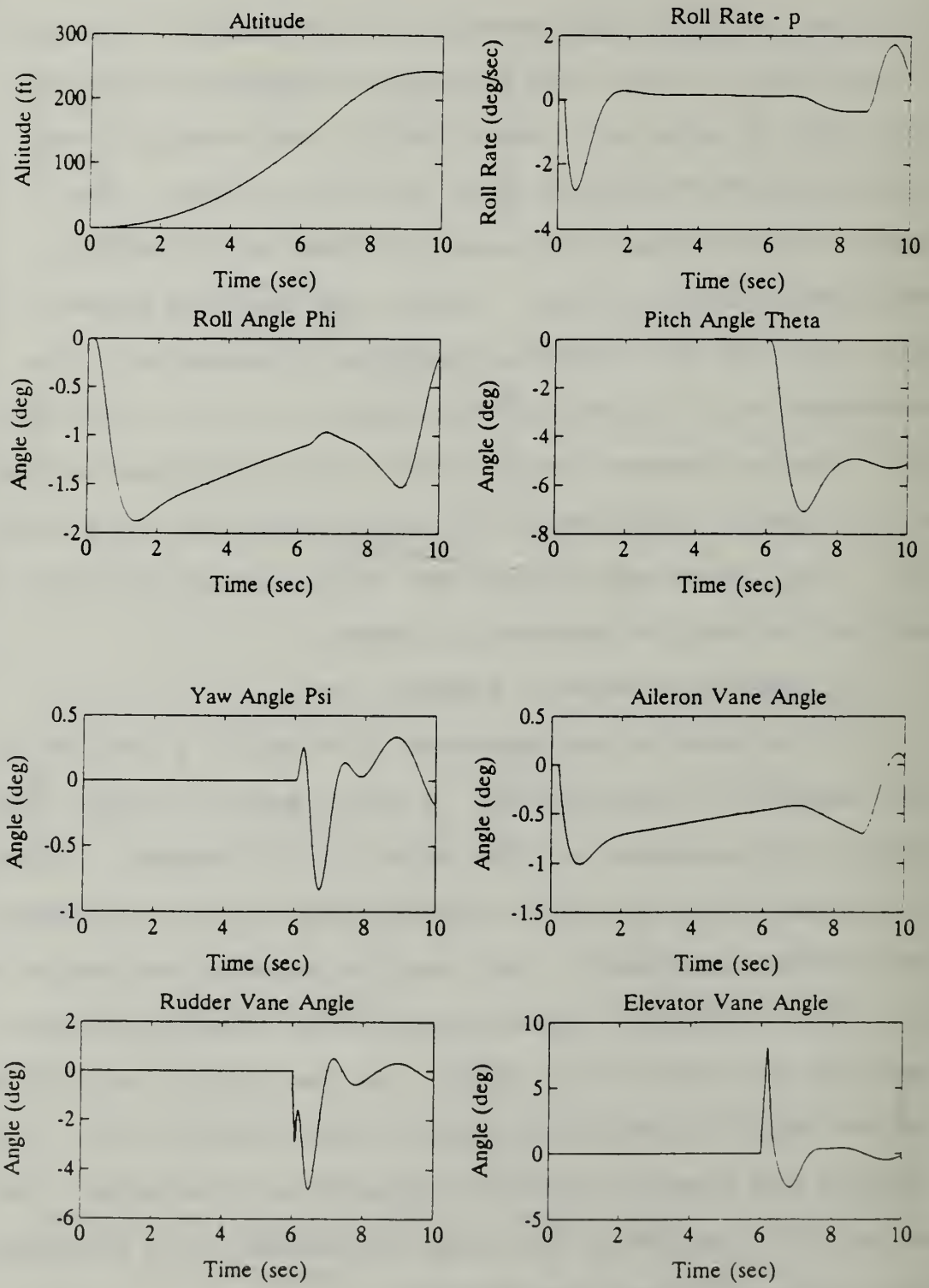


value of approximately 80 feet. The large overshoot for the altitude is due to the momentum of the vehicle. Climbing at a high rate of climb, the vehicle is expected to exhibit this type of behavior. Additionally, the control system is controlling the altitude rate, not the altitude. The operator would have to account for this in his control adjustments during flight. Again, the coupling between the roll axis and the engine acceleration is observed. The commanded roll rate drives the vehicle to an roll angle of 18°. The two degree error is due to the minus two degrees of roll angle caused by the reactive torque to the altitude rate. The increased aileron vane displacement in order to perform the desired maneuver is shown.

3. Simulation Three - Figure 4.18

The vehicle was commanded to climb at a rate of 50 feet/second for four seconds. A pitch angle of minus five degrees was commanded at time equal to six seconds. Figure 4.18 illustrates that the increased rate of climb increases the altitude overshoot. The coupling between the engine and roll axis is present. Additionally, the coupling between the pitch and yaw axis is shown. As the vehicle is pitched, the yaw angle is perturbed. Note, the controller is limiting the coupling between the pitch and yaw axis. The yaw angle is perturbed less than one degree for a commanded pitch angle of minus five degrees.

Figure 4.18 Nonlinear Simulation Three



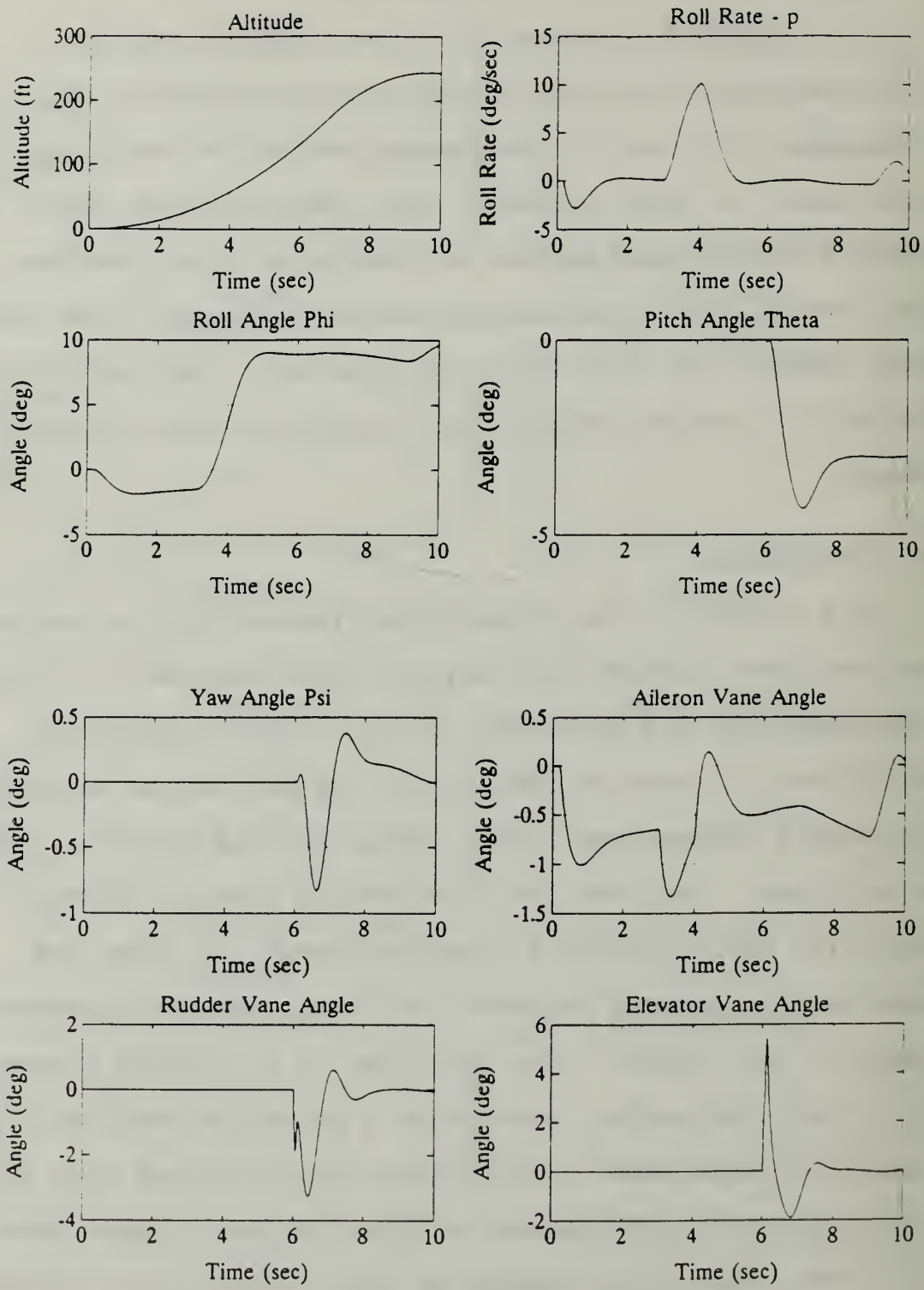
4. Simulation Four - Figure 4.19

Figure 4.19 shows the system response for an altitude rate of 50 feet/second for four seconds. The commanded roll rate is ten degrees/second for one second at time equal to three seconds. The commanded pitch angle is minus three degrees applied at time equal to six seconds. The control vanes displacement angles all remain less than four degrees for this series of maneuvers. The controller is able to provide satisfactory results for the nonlinear model.

F. CONCLUSION

The controller has demonstrated the ability to control the nonlinear system with varying inputs applied. It can be concluded that the robustness of each control system is sufficient in order to account for the difference in system parameters between the linear models and the nonlinear model of Archytas. The fact that the control systems maintain stability while directing commanded inputs to steer and maneuver the vehicle supports the linear modeling approach taken in this thesis. The next step in the design process is to test the control systems on a prototype vehicle in a controlled experiment. Such a test was performed with the roll rate control system and an Archytas prototype mounted on a test stand. The results of this test will be discussed in Chapter V.

Figure 4.19 Nonlinear Simulation Four



V. CONCLUSIONS

A. ROLL RATE CONTROL SYSTEM FIELD TEST

Figure 5.1 shows the Archytas prototype (minus the wings and canard) mounted on a test stand that allows the vehicle to spin about the longitudinal, or x axis. This configuration was used during the testing of the roll rate controller. The testing was accomplished in a qualitative manner; no empirical data was collected, the object of the test being to validate the roll rate controller design. The test consisted of three parts:

1. The ability of the roll rate controller to eliminate the rotational velocity imparted to Archytas from the reactive torques applied to the roll axis as the engine speed is varied (decoupling).
2. The ability of the roll rate controller to eliminate the rotational velocity imparted to Archytas from the effect of cross-winds (disturbance rejection).
3. The ability of the roll rate controller to allow the operator to impart or terminate a rotational velocity about the x axis (tracking commanded inputs).

Part one was accomplished by varying the engine speed from idle to maximum rpm with the commanded roll rate equal to zero. Each time the engine speed was varied, the rotational velocity of the vehicle was eliminated by the roll rate controller. It was observed that the rotational velocity was eliminated within approximately two seconds.

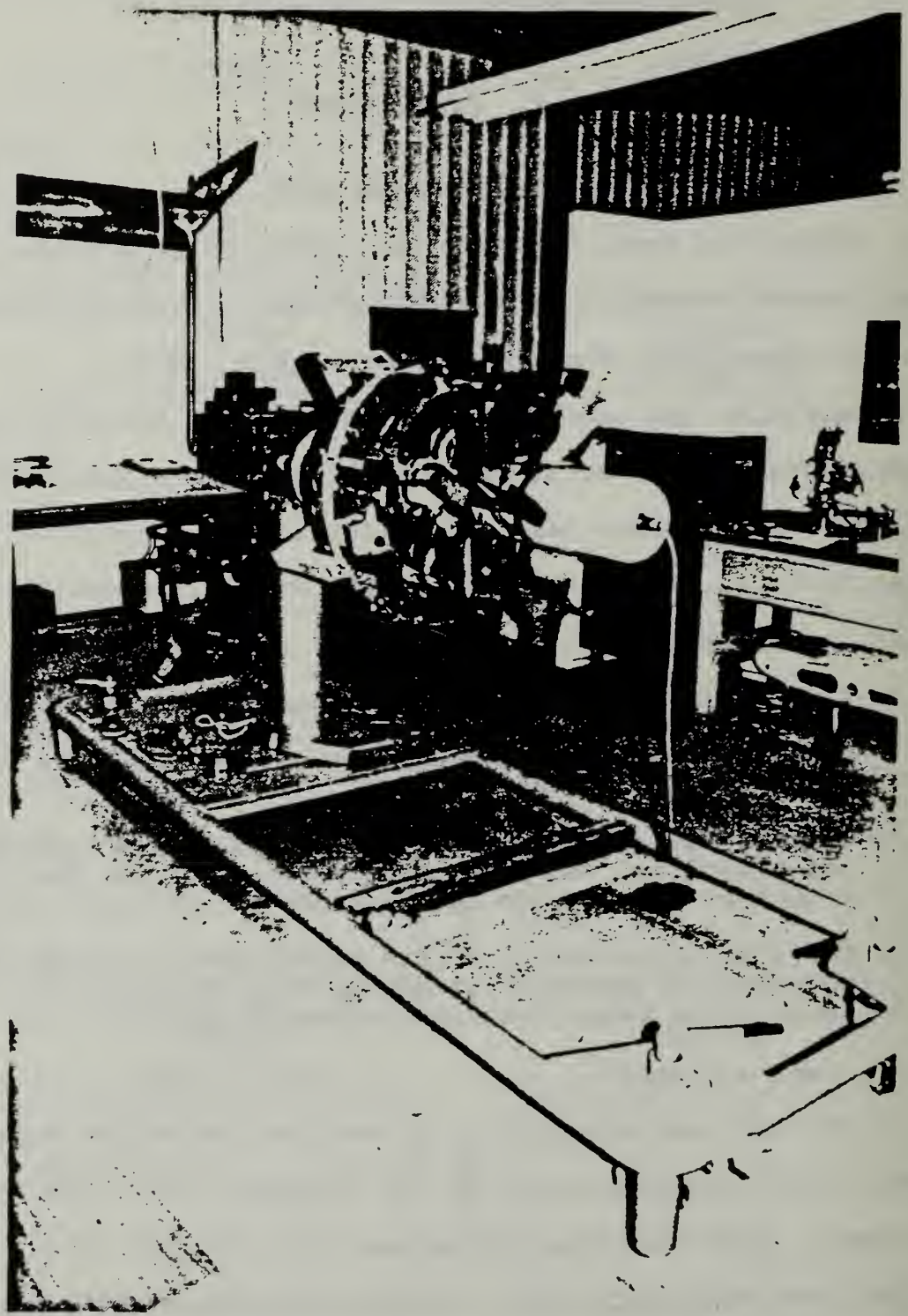


Figure 5.1 Archytas Prototype Mounted on the Test Stand

This response is consistent with both the linear and nonlinear simulations.

Part two was accomplished by setting the engine speed at constant values ranging from idle to maximum rpm with the commanded roll rate equal to zero. A disturbance about the x axis was imparted to Archytas by pushing the test stand mounting bracket connected to the vehicle. The effect of pushing the connecting bracket is analogous to a rotational velocity imparted by cross-winds. It was observed that the rotational velocity was eliminated within approximately two seconds. This response is consistent with both the linear and nonlinear simulations.

Part three was accomplished by inputting a commanded roll rate while the engine speed was held constant and also varied. It was observed that the operator could position Archytas at a desired angle under either test condition. Additionally, disturbances were imparted to Archytas during these tests. The roll rate controller proved robust enough to eliminate the disturbances and allow the operator to position the vehicle. This response is consistent with both the linear and nonlinear simulations.

Based on the computer simulations, both linear and nonlinear, and the results of the above tests, it is concluded that the roll rate controller design is valid.

B. FUTURE RESEARCH

The development of the Archytas control systems was accomplished using the data from the AROD as first approximations. Future research will include the computation of Archytas specific data through empirical methods or obtaining the data by wind tunnel testing. This would allow for a better model of the Archytas to be developed.

The second order dynamical model of the servos was obtained with the vanes mounted on the servos with zero downwash from the propeller. Future research should include modeling of the servos with the engine at hover rpm and the vanes positioned within the downwash.

The effects of inclusion of the vane position angles in the control systems could be evaluated further. In particular, the effect the vane angle position feedback would have in the reduction of the overshoot in the pitch and yaw angle controller should be investigated.

APPENDIX A

MATLAB SIMULATION PROGRAMS

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   PROGRAM NAME:   nonlin.m                                              %
%                                                                         %
%   DESCRIPTION:                                                          %
%   This program will simulate the Archytas nonlinear model%
%   with the linear controllers.  The Control Laws are the %
%   Steady-state Linear Quadratic Regulator solutions with %
%   the necessary modifications to the performance measure.%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear
%  Call the program noninp.m to enter the desired roll rate,
%  alttitude rate, pitch and yaw angles.
noninp
%  Call the program in_cond.m to load the initial conditions
and %  constants for the simulation.
in_cond
%  Call the program n_l_inp.m to load the wind tunnel data.
n_l_inp
%  Loop for simulation
for k=1:num_of_steps1

%  Compute the speed of the propeller
speed(k)  = speed_hover + del_rpm(k);
%  Compute the thrust supplied by the engine to the vehicle
thrust(k) = thrust_hover + Xrpm * del_rpm(k);
%  Check thrust limits
if thrust(k) < thrust_min
   thrust(k) = thrust_min;
elseif thrust(k) > thrust_max
   thrust(k) = thrust_max;
end

velocity_tip(k) = speed_hover + del_rpm(k);
del_tip(k) = velocity_tip(k) - speed_hover;
tip_del(k) = velocity_tip(k) - speed(k);
stad(k)    = tip_del(k);

%  Compute vehicle velocity and angle of attack total
aoatot(k) = alpha_max;
velocity_tot(k)=sqrt(u(k)^2+v(k)^2+w(k)^2);
if velocity_tot(k) > vaero
     vwterm = sqrt(v(k)^2 + w(k)^2);
     aoatot(k) = asin(vwterm/velocity_tot(k));
       if aoatot(k) < alpha_min
```

```
         aoatot(k) = alpha_min;
      elseif aoatot(k) > alpha_max
         aoatot(k) = alpha_max;
      end
end

veq(k) = table1(veq_table,aoatot(k));
newveq(k) = sqrt(weight_ratio)*veq(k);
rhova(k)  = rho * newveq(k) * pi;
velocity_tip_2(k) = 2.0/velocity_tip(k);
velocity_delta(k) = velocity_tot(k) - newveq(k);

%  This section assumes that no aerodynamic forces and moments
%  exist
if velocity_tot(k) <= vaero

      fax          = thrust(k) * gravity/weight;
      fay          = 0.0;
      faz          = 0.0;
      mrx          = 0.0;
      mpy          = 0.0;
      myz          = 0.0;
      pitch_trim   = 0.0;
      yaw_trim     = 0.0;
      pitch_factr  = 1.0;
      yaw_factr    = 1.0;

elseif velocity_tot(k) > vaero
      if aoatot(k) <= alpha_min

         fax          = thrust(k) * gravity/weight;
         fay          = 0.0;
         faz          = 0.0;
         mrx          = 0.0;
         mpy          = 0.0;
         myz          = 0.0;
         pitch_trim   = 0.0;
         yaw_trim     = 0.0;
         pitch_factr  = 1.0;
         yaw_factr    = 1.0;
      end

else

%  This section computes the aerodynamic forces and moments

      lleq    = weight_ratio * leq;
      ddeq    = weight_ratio * deq;
      sseq    = weight_ratio * seqq;

      req     = table1(req_table,aoatot(k));
```

```
peq     = table1(peq_table,aoatot(k));
yeq     = table1(yeq_table,aoatot(k))

rreq    = weight_ratio * req;
ppeq    = weigth_ratio * peq;
yyeq    = weight_ratio * yeq;

cldel   = table1(cldel_table,aoatot(k));
cddel   = table1(cddel_table,aoatot(k));
csdel   = table1(csdel_table,aoatot(k));
ldel = velocity_tip_2(k)*lleq-rhova*weight_ratio*cldel;
ddel = velocity_tip_2(k)*ddeq-rhova*weight_ratio*cddel;
sdel = velocity_tip_2(k)*sseq-rhova*weight_ratio*csdel;

crdel   = table1(crdel_table,aoatot(k));
cpdel   = table1(cpdel_table,aoatot(k));
cydel   = table1(cydel_table,aoatot(k));

rdel = velocity_tip_2(k)*rreq-rhova*weight_ratio*crdel;
pdel = velocity_tip_2(k)*ppeq-rhova*weight_ratio*cpdel;
ydel = velocity_tip_2(k)*yyeq-rhova*weight_ratio*cydel;

lslope = table1(lslope_table,aoatot(k));
dslope = table1(dslope_table,aoatot(k));
sslope = table1(sslope_table,aoatot(k));

ffl     = lleq+ldel*del_tip(k)+lslope*weight_ratio*
          velocity_delta(k);
ffd     = ddeq+ddel*del_tip(k)+dslope*weight_ratio*
          velocity_delta(k);
ffs     = sseq+sdel*del_tip(k)+sslope*weight_ratio*
          velocity_delta(k);

rslope = table1(rslope_table,aoatot(k));
pslope = table1(pslope_table,aoatot(k));
yslope = table1(yslope_table,aoatot(k));

mr      = rreq+rdel*del_tip(k)+rslope*weight_ratio*
          velocity_delta(k);
mp      = ppeq+pdel*del_tip(k)+pslope*weight_ratio*
          velocity_delta(k);
my      = yyeq+ydel*del_tip(k)+yslope*weight_ratio*
          velocity_delta(k);

fs      = ffs * gravity/weight;
fl      = ffl * gravity/weight;
fd      = ffd * gravity/weight;

delta   = atan(v(k)/w(k));

fax     = fl * sin(aoatot(k)) - fd * cos(aoatot(k));
```

```
      fayz   =-fl * cos(aoatot(k)) - fd * sin(aoatot(k));
      fay    = fayz * sin(delta);
      faz    = fayz * cos(delta);

      mrx    = my * sin(aoatot(k)) - mr * cos(aoatot(k));
      mpyyz  =-my * cos(aoatot(k)) - mr * sin(aoatot(k));
      mpy    = mpyyz * sin(delta);
      myz    = mpyyz * cos(delta);

      velocity_trim =table1(velocity_trim_table,aoatot(k1));
      vaneff = table1(vaneff_table,aoatot(k));


      pitch_trim  = velocity_trim * cos(delta);
      yaw_trim    =-velocity_trim * sin(delta);
      pitch_factr = vaneff;
      yaw_factr   = vaneff;

end

lat(k) = ((rscale*mrx*gravity*144.0)/Ixx)+(La*del_a(k))-
         (velocity_delta(k)^2 /Ixx) * prop_torq;

mat(k) = ((mpy*gravity*144.0)/Iyy)+(Me*pitch_factr*
         (del_e(k)-pitch_trim));
nat(k) = ((myz*gravity*144.0)/Izz)+(Nr*yaw_factr *
         (del_r(k)-yaw_trim));

%  Begin Differential Equations of Motion

%  Altitude Differential Equations
temp1(k)=u(k)*cos(theta(k))*cos(psi(k));
temp2(k)=v(k)*(sin(phi(k))*sin(theta(k))*cos(psi(k))-
         cos(phi(k))*sin(psi(k)));
temp3(k)=w(k)*(cos(phi(k))*sin(theta(k))*cos(psi(k))+
         sin(phi(k))*sin(psi(k)));
u_earth(k)= temp1(k)+temp2(k)+temp3(k);

%  Heading Angle Differential Equations
temp5(k)=p(k)*cos(theta(k))*cos(psi(k));
temp6(k)=q(k)*(sin(phi(k))*sin(theta(k))*cos(psi(k))-
         cos(phi(k))*sin(psi(k)));
temp7(k)=r(k)*(cos(phi(k))*sin(theta(k))*cos(psi(k))
         +sin(phi(k))*sin(psi(k)));
hdg_dot(k)=temp5(k)+temp6(k)+temp7(k);

%  Euler Angle Differential Equations: Pitch, Yaw, and Roll
theta_dot(k)=q(k)*cos(phi(k))-r(k)*sin(phi(k));
phi_dot=p(k)+(q(k)*sin(phi(k))+r(k)*cos(psi(k)))*
        tan(theta(k));
```

```
psi_dot(k)=(q(k)*sin(phi(k))+r(k)*cos(phi(k)))/
           cos(theta(k));

%  Vehicle Pitch, Yaw and Roll Rate Differential Equations WRT
%  Body Axes
p_dot(k) = (bca*q(k)*r(k))-(Iralph*stad(k))+lat(k);
q_dot(k) = (cab*p(k)*r(k))-(Irx*r(k)*speed(k)/Iyy)+mat(k);
r_dot(k) = (abc*p(k)*q(k))+(Irx*q(k)*speed(k)/Izz)+nat(k);

% Vehicle Velocity Differential Equations WRT Body Axes
u_dot(k)=((v(k)*r(k))-(w(k)*q(k)))-gravity*cos(theta(k))*
         cos(psi(k))+fax;
v_dot(k)=((w(k)*p(k))-(u(k)*r(k)))+gravity*cos(theta(k))*
         sin(psi(k))+fay;
w_dot(k)=((u(k)*q(k)) -(v(k)*p(k)))-gravity*
         sin(theta(k))+faz;


%  Define Error Variable in Pitch and Yaw
E_theta(k)=theta_com(k)-(theta(k)+noise(k));
E_psi(k)=psi_com(k)-(psi(k)+noise(k));

%  Define Error Variable in Roll
E_p(k)=p_com(k)-(p(k)+noise(k));

%  Define Error Variable in Altitude
E_h_dot(k)=h_dot_com(k)-(u(k)+noise(k));

%  Integrate Errors
E_thetain(k+1)=E_thetain(k)+Ts*E_theta(k);
E_psiin(k+1)=E_psiin(k)+Ts*E_psi(k);
E_pin(k+1)=E_pin(k)+Ts*E_p(k);
E_h_dotin(k+1)=E_h_dotin(k)+Ts*E_h_dot(k);

%  Define Control System Gains
%  Roll Rate Control Gains
K1=[ 0.4376 -0.2027];

%  Pitch and Yaw Angle Control Gains
K2=[ 2.2741  1.8933 -0.6059 0.6390 0.5320  0.0024;
    -0.6390 -0.5320 -0.0024 2.2741 1.8933 -0.6059];

%  Altitude Rate Control Gains
K3=[-0.3060 0.2003 0.0038];

%  Calculate the Aileron (roll) Servo Control Input
Ua(k)=-K1*[E_pin(k+1);(p(k)+noise(k))];

%  Calculate the Elevator (pitch) Servo Control Input
Ue(k)=-K2(1,:)*[E_thetain(k+1);E_theta(k);q(k);E_psiin(k+1);
               E_psi(k);r(k)];
```

```
%  Calculate the Rudder (yaw) Servo Control Input
Ur(k)=-K2(2,:)*[E_thetain(k+1);E_theta(k);q(k);E_psiin(k+1);
                E_psi(k);r(k)];

%  Calculate the Throttle (altitude) Servo Control Input
Ut(k)=-K3*[E_h_dotin(k+1);u(k);del_rpm(k)];

%  Begin Integration of Equations of Motion
del_a_dot_dot(k)=-H1*del_a_dot(k)-H2*del_a(k)+H2*Ua(k);
del_e_dot_dot(k)=-H1*del_e_dot(k)-H2*del_e(k)+H2*Ue(k);
del_r_dot_dot(k)=-H1*del_r_dot(k)-H2*del_r(k)+H2*Ur(k);
del_t_dot_dot(k)=-H1*del_t_dot(k)-H2*del_t(k)+H2*Ut(k);

%Roll, pitch, yaw
p(k+1)=p(k)+Ts*p_dot(k);
q(k+1)=q(k)+Ts*q_dot(k);
r(k+1)=r(k)+Ts*r_dot(k);

%Velocities
u(k+1)=u(k)+Ts*u_dot(k);
v(k+1)=v(k)+Ts*v_dot(k);
w(k+1)=w(k)+Ts*w_dot(k);

%Euler angles
theta(k+1)=theta(k)+Ts*theta_dot(k);
phi(k+1)=phi(k)+Ts*phi_dot(k);
psi(k+1)=psi(k)+Ts*psi_dot(k);

%Servos
del_a_dot(k+1)=del_a_dot(k)+Ts*del_a_dot_dot(k);
if del_a_dot(k+1) > rmax
   del_a_dot(k+1) = rmax;
elseif del_a_dot(k+1) < -rmax
   del_a_dot(k+1) = -rmax;
end

del_e_dot(k+1)=del_e_dot(k)+Ts*del_e_dot_dot(k);
if del_e_dot(k+1) > rmax
   del_e_dot(k+1) = rmax;
elseif del_e_dot(k+1) < -rmax
   del_e_dot(k+1) = -rmax;
end

del_r_dot(k+1)=del_r_dot(k)+Ts*del_r_dot_dot(k);
if del_r_dot(k+1) > rmax
   del_r_dot(k+1) = rmax;
elseif del_r_dot(k+1) < -rmax
   del_r_dot(k+1) = -rmax;
end
```

```
del_t_dot(k+1)=del_t_dot(k)+Ts*del_t_dot_dot(k);
if del_t_dot(k+1) > rmaxt
   del_t_dot(k+1) = rmaxt;
elseif del_t_dot(k+1) < -rmaxt
   del_t_dot(k+1) = -rmaxt;
end

%  Engine
del_rpm_dot(k+1)=-w_e*del_rpm(k)+Ke*w_e*del_t(k);

del_a(k+1)=del_a(k)+Ts*del_a_dot(k);

del_e(k+1)=del_e(k)+Ts*del_e_dot(k);
if del_e(k+1) > maxdfl
   del_e(k+1) = maxdfl;
elseif del_e(k+1) < -maxdfl
   del_e(k+1) = -maxdfl;
end

del_r(k+1)=del_r(k)+Ts*del_r_dot(k);
if del_r(k+1) > maxdfl
   del_r(k+1) = maxdfl;
elseif del_r(k+1) < -maxdfl
   del_r(k+1) = -maxdfl;
end

del_t(k+1)=del_t(k)+Ts*del_t_dot(k);
if del_t(k+1) > tamax
   del_t(k+1) = tamax;
elseif del_t(k+1) < -tamax
   del_t(k+1) = -tamax;
end

%  Engine
del_rpm(k+1)=del_rpm(k)+Ts*del_rpm_dot(k);

%  Altitude
alt(k+1)=alt(k)+Ts*u_earth(k);

%  Heading Angle
heading(k+1)=heading(k)+Ts*hdg_dot(k);

t(k)=Ts*(k-1);
%  End of simulation loop
end

t(k+1)=Ts*k;
u_earth(k+1)=u_earth(k);
```

```
% Plot the Results of the Simulation
subplot(221),plot(t,alt),grid,xlabel('Time
(sec)'),ylabel('Altitude (ft)');
title('Altitude');
subplot(222),plot(t,p*57.2958),grid,xlabel('Time
(sec)'),ylabel('Roll Rate (deg/sec)');
title('Roll Rate - p');

subplot(223),plot(t,phi*57.2958),grid,xlabel('Time
(sec)'),ylabel('Angle (deg)');
title('Roll Angle Phi');
subplot(224),plot(t,theta*57.2958),grid,xlabel('Time
(sec)'),ylabel('Angle (deg)');
title('Pitch Angle Theta');
meta non1
subplot(111);

subplot(221),plot(t,psi*57.2958),grid,xlabel('Time
(sec)'),ylabel('Angle (deg)');
title('Yaw Angle Psi');
subplot(222),plot(t,del_a*57.2958),grid,xlabel('Time
(sec)'),ylabel('Angle (deg)');
title('Aileron Vane Angle');

subplot(223),plot(t,del_r*57.2958),grid,xlabel('Time
(sec)'),ylabel('Angle (deg)');
title('Rudder Vane Angle');
subplot(224),plot(t,del_e*57.2958),grid,xlabel('Time
(sec)'),ylabel('Angle (deg)');
title('Elevator Vane Angle');
meta non2
subplot(111);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   PROGRAM NAME: noninp.m                                 %
%                                                          %
%   DESCRIPTION:                                           %
%   This program prompts the user for the desired          %
%   simulation inputs.                                     %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Prompt the user for the desired simulation time.
disp(' Enter the simulation time in seconds.');
simulation_time = input('>>>> ');

% Prompt the user to enter to desired sampling time - Ts.
disp(' Enter the desired sampling time - Ts.');
```

```
Ts = input('>>>> ');

%  Compute the number of simulation steps based on the
desired simulation time.
num_of_steps1   = round(simulation_time/Ts);

%  Prompt the user for the length of the altitude rate step
input time.
disp(' Enter the length of the altitude rate input in
seconds.');
step_input_time1      = input('>>>> ');

%  If step_input entered is greater than simulation_time
entered, set them equal.
if step_input_time1 > simulation_time
   step_input_time1 = simulation_time;
end

if step_input_time1 ~= 0
%  Prompt the user for the desired magnitude of the step
%  input in # of deg/sec.
   disp(' Enter the magnitude of the altitude rate input in
 feet/sec.')
   h_dot = input('>>>> ');
elseif step_input_time1 == 0
   h_dot = 0.0;
end

%  Compute the number of step input steps based on the
%  desired step input time.
num_of_steps2   = step_input_time1/Ts;
num_of_steps2   = round(num_of_steps2);

%  Compute the number of zeros to be padded to the input
%  based on the length simulation time and length of the
%  desired step input time.

num_of_zeros1 = num_of_steps1 - num_of_steps2;

%  Define the commanded input vector
h_dot_com = [h_dot * ones(1,num_of_steps2)
zeros(1,num_of_zeros1)];

%  Prompt the user for whether he wants to include a roll
rate in the simulation.
ans = input('Do you want to include a roll rate in the
simulation? y/n [y]: ','s');
if isempty(ans)
   ans ='y';
end
if ans =='y'
```

```
   %  Prompt the user for the start time for the roll rate
step input.
   disp(' Enter the start time for the roll rate input.');
   start_time1 = input('>>>> ');

   %  Prompt the user for the length of the roll rate input.
   disp(' Enter the length of the roll rate imput in
seconds.');
   length_roll_rate = input('>>>> ');

   %  Compute the number of zeros to pad front of the
   %  command vector.
   num_of_zeros2 = start_time1/Ts;
   num_of_zeros2 = round(num_of_zeros2);

   %  Computer the number of step input steps and number of
   %  zeros to pad back of the command vector

   num_of_steps3 = length_roll_rate/Ts;
   num_of_steps3 = round(num_of_steps3);
   num_of_zeros3 = num_of_steps1-
                   round((start_time1+length_roll_rate)/Ts);

   %  Prompt the user for the desired magnitude of the roll
   %  rate input in # deg/se.
   disp('Enter the magnitude of the roll rate in
         degrees/sec.');
   p1 = input('>>>> ');
   %  Define the commanded input vector
   p_com = [zeros(1,num_of_zeros2) p1 * 0.017453 *
            ones(1,num_of_steps3) zeros(1,num_of_zeros3)];
   else
   p_com = [zeros(1,num_of_steps1)];
end

%  Prompt the user for whether he wants to include a pitch
%  angle in the simulation.
ans = input('Do you want to include a pitch angle in the
      simulation? y/n [y]: ','s');
if isempty(ans)
   ans = 'y';
end
if ans =='y'
   %  Prompt the user for the start time for the pitch angle
   %  command.
   disp(' Enter the time at which you desire to input the
         pitch angle.')
   start_time2 = input('>>>> ');
   %  Compute the number of zeros to pad front of the
   %  command vector
   num_of_zeros4 = round(start_time2/Ts);
```

```
   %  Compute the number of ones to multiply the desired
   %  angle by to create the command vector.
   number_of_ones1 = num_of_steps1 - round(start_time2/Ts);
   %  Prompt the user for the desired pitch angle in
   %  degrees.
   disp(' Enter the desired pitch angle in degrees.');
   theta_com = input('>>>> ');
   theta_com = [zeros(1,num_of_zeros4) theta_com * 0.017453
            * ones(1,number_of_ones1)];
else
   theta_com = [zeros(1,num_of_steps1)];
end

%  Prompt the user for whether he wants to include a yaw
%  angle in the simulation.
ans = input('Do you want to include a yaw angle in the
      simulation? y/n [y]: ','s');
if isempty(ans)
   ans = 'y';
end
if ans =='y'
   %  Prompt the user for the start time for the yaw angle
   %  command.
   disp(' Enter the time at which you desire to input the
       yaw angle.')
   start_time2 = input('>>>> ');
   %  Compute the number of zeros to pad front of the
   %  command vector
   num_of_zeros4 = round(start_time2/Ts);
   %  Compute the number of ones to multiply the desired
   %  angle by to create the command vector.
   number_of_ones1 = num_of_steps1 - round(start_time2/Ts);
   %  Prompt the user for the desired pitch angle in
   %  degrees.
   disp(' Enter the desired yaw angle in degrees.');
   psi_com = input('>>>> ');
   psi_com = [zeros(1,num_of_zeros4) psi_com * 0.017453 *
             ones(1,number_of_ones1)];
else
   psi_com = [zeros(1,num_of_steps1)];
end

%  Prompt the user for whether he wants to add sensor noise
%  to the simulation.
ans = input('Do you want to include sensor noise in the
      simulation? y/n [y]: ','s');
if isempty(ans)
   ans ='y';
end

if ans =='y'
```

```
   %  Define the random vector to represent the measurement
   %  noise.
   %  Assume the accuracy is +/- 1.745e-04.
   rand('uniform');
   noise = rand(1,num_of_steps1);
   vard  = (1/12) * (1.745e-04)^2;    %  Desired variance
   A     = sqrt(vard/cov(noise));     %  Scalar multiplier to
                                      %  change the variance
   noise = A .* noise;
   %  Adjust the mean to zero
   noise = noise - mean(noise);
else
   noise = [zeros(1,num_of_steps1)];
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   PROGRAM NAME:   in_cond.m                              %
%                                                          %
%   DESCRIPTION:                                           %
%   This program inputs the initial conditions and         %
%   constant values used in nonlin.m                       %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%  CONSTANTS
vaero=0.5; speed_max=859.0; speed_min=649.0;
prop_torq=0.0729; thrust_min=35.0; thrust_max=105.0;
tamax=100.0; rmaxt=100.0;
maxdfl=0.5236; rmax=0.87266;
La=-21.04; Me=-9.01; Nr=-11.40;
H1=71.1; H2=2745.8;
w_e=2.0; Ke=837.758; Xrpm=0.2122; speed_hover=712.0943;
alpha_min=0.174533; alpha_max=1.570796;
gravity=32.174; pi=3.1415962; rad_to_deg=180.0/pi;
Ixx=6908.4; Iyy=22944.64; Izz=21515.64; Irx=41.62;
bca=0.20685; cab=0.63663; abc=-0.74533; Iralph=0.00916;
weight=85.0; thrust_hover=85.0; velocity_tip=722.5663;
rho=0.00192; weight_ratio=1.0;
leq=85.0; deq=0.0; seqq=0.0; sslope=0.0801; rscale=0.0;

%  INITIAL CONDITIONS


p(1)=0.0;
q(1)=0.0;
r(1)=0.0;
```

```
u(1)=0.0;
v(1)=0.0;
w(1)=0.0;

phi(1)  =0.0;
psi(1)  =0.0;
theta(1)=0.0;

alt(1)  =0.0;
dist(1) =0.0;
heading(1) =0.0;

E_thetain(1)  =0.0;
E_psiin(1)    =0.0;
E_pin(1)      =0.0;
E_h_dotin(1)  =0.0;

del_a(1)  =0.0;
del_e(1)  =0.0;
del_r(1)  =0.0;
del_t(1)  =0.0;
del_rpm(1)=0.0;

del_a_dot(1)=0.0;
del_e_dot(1)=0.0;
del_r_dot(1)=0.0;
del_t_dot(1)=0.0;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   PROGRAM NAME:   n_l_inp.m                                %
%                                                            %
%   DESCRIPTION:                                             %
%   This file inputs the wind tunnel data into the           %
%   MATLAB wordspace                                         %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

aoatot1=[0.174533 0.349066 0.523599 0.785398 0.872665...
         0.959931 1.047198 1.134464 1.221730 1.308997...
         1.570796];

CRdel1 =[0.00 0.00 0.03 0.06 0.06
         0.07 0.08 0.07 0.07 0.07 0.05];
Rslope1=[0.00 0.01 0.07 0.14 0.16
         0.18 0.18 0.18 0.17 0.18 0.20];
```

```
Req1   =[2.16 4.84 6.66 7.23 7.76
         8.66 8.92 6.89 6.66 5.68 0.00];

CPdel1 =[-0.08 -0.09 -0.08 -0.02 -0.00
          0.03 0.07 0.12 0.08 0.12 0.00];
Pslope1=[-0.29 -0.24 -0.21 -0.04 -0.01
          0.08 0.17 0.29 0.19 0.31 0.40];
Peq1   =[-5.06 2.24 7.46 15.32 16.65 17.12
         17.37 17.21 18.03 16.10 0.00];

CYdel1 =[0.03 0.04 0.04 0.03 0.03
         0.03 0.03 0.01 0.01 0.01 0.00];
Yslope1=[0.12 0.12 0.11 0.08 0.07
         0.06 0.06 0.02 0.02 0.02 0.00];
Yeq1   =[4.54 6.36 6.82 3.13 3.05
         3.73 3.17 2.34 2.34 1.81 1.00];

VanEff1=[1.50 1.45 1.40 1.35 1.30
         1.25 1.20 1.15 1.10 1.05 1.00];
Veq1   =[172.4 115.4 87.65 62.85 55.45
         48.00 41.2234.68 28.98 22.37 0.0];

CLdel1 =[0.16 0.28 0.32 0.29 0.30 0.26
         0.23 0.19 0.20 0.16 0.00];
Lslope1=[0.60 0.81 0.82 0.73 0.74 0.64
         0.56 0.46 0.50 0.40 0.00];

CDdel1 =[0.44 0.46 0.46 0.48 0.50 0.50
         0.50 0.50 0.54 0.52 0.50];
Dslope1=[1.67 1.32 1.19 1.20 1.23 1.22
         1.24 1.23 1.34 1.32 1.25];

CSdel1 =[-0.01 -0.02 -0.03 -0.01 -0.01
         -0.02 0.01 0.00 0.02 0.04 0.06];
Vtrim1 =[0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0];

crdel_table =[aoatot1'CRdel1'];
rslope_table=[aoatot1'Rslope1'];
req_table   =[aoatot1' Req1'];

cpdel_table =[aoatot1' CPdel1'];
pslope_table=[aoatot1' Pslope1'];
peq_table   =[aoatot1' Peq1'];

cydel_table =[aoatot1' CYdel1'];
yslope_table=[aoatot1' Yslope1'];
yeq_table   =[aoatot1' Yeq1'];

vanEff_table=[aoatot1' VanEff1'];
veq_table   =[aoatot1' Veq1'];
```

```
cldel_table =[aoatot1' CLdel1'];
lslope_table=[aoatot1' Lslope1'];

cddel_table =[aoatot1' CDdel1'];
dslope_table=[aoatot1' Dslope1'];

csdel_table =[aoatot1' CSdel1'];
velocity_trim_table =[aoatot1'  Vtrim1'];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

APPENDIX B CONTROL SERVOS

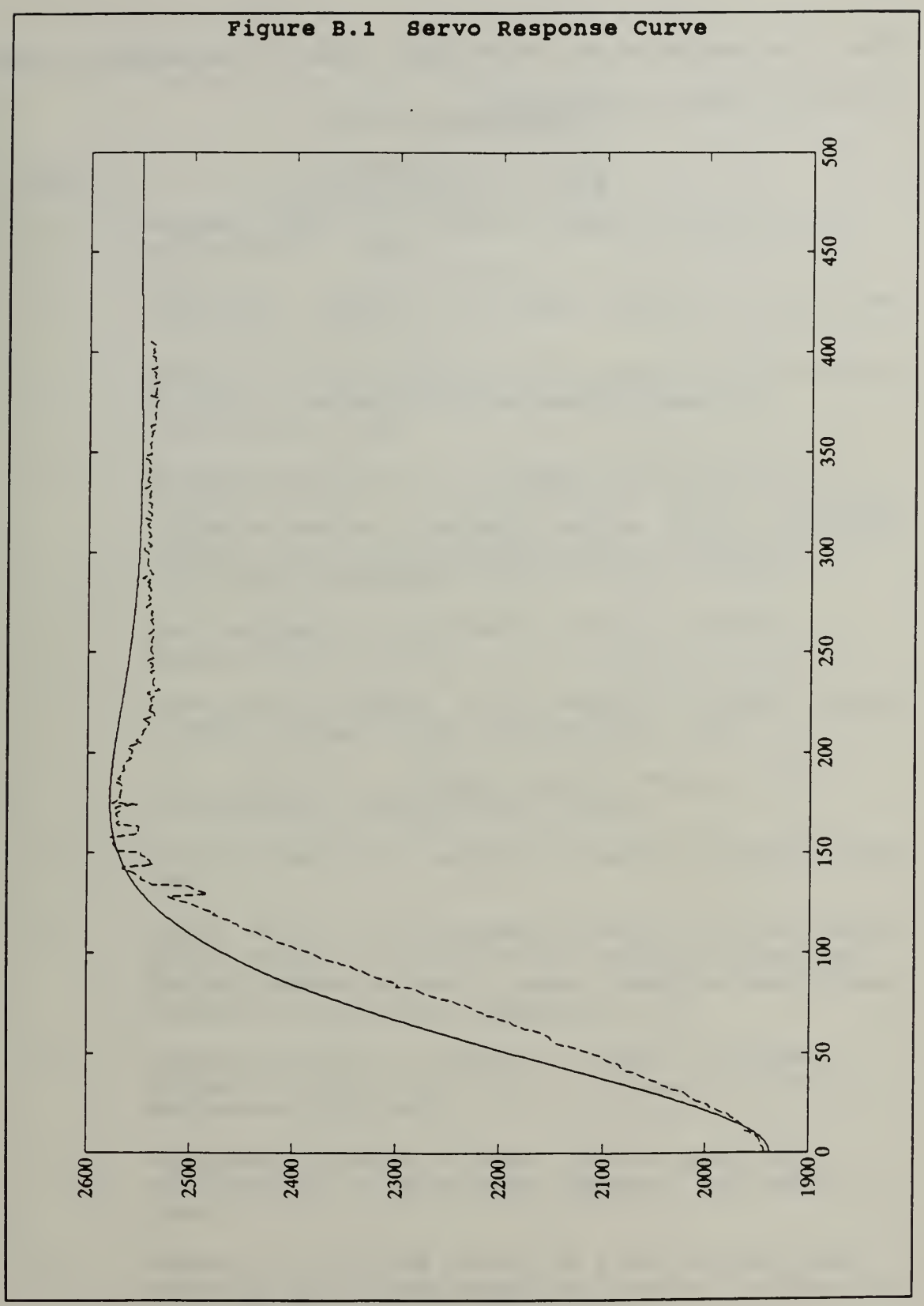
The control servos incorporated into Archytas are position-commanded, quarter-scale model airplane servos. Five of these servos are used on Archytas: four to drive the aerodynamic control surfaces and one to drive the engine throttle. These servos can be modeled as second-order dynamical systems:

$$\frac{C(s)}{R(s)} = \frac{\omega_n^2}{s^2+2\zeta\omega_n+\omega_n^2}, \qquad \text{(B.1)}$$

where C(s) is the output and R(s) is the input. ζ and ω_n are referred to as the damping ratio and the natural frequency, respectively.

The values for ζ and ω_n were determined by experiment. The control vanes were mounted on the servos and position data was obtained for a range of step input commands. The input commands ranged from three to thirty degrees of vane deflection. By plotting the servo angle response data versus time, a series of step input response curves was developed. Using MATLAB, trial and error was used to fit a second order prototype system to the step response curves of the servos. Figure B.1 shows the "best fit" model. This model is defined by ζ=0.707 and ω_n=52.4 radians/second.

Figure B.1 Servo Response Curve



Thus, the second order servo model used in the design of the Archytas control systems is:

$$\frac{C(s)}{R(s)} = \frac{2745.8}{s^2+74.1s+2745.8} \qquad \textbf{(B.2)}$$

LIST OF REFERENCES

1. Nelson, Robert C., *Flight Stability and Control*, McGraw-Hill, 1989.

2. Etkin, B., *Dynamics of Flight*, 2nd ed., John Wiley and Sons., 1982.

3. Roskam, Jan, *Airplane Flight Dynamics and Automatic Controls*, Roskam Aviation and Engineering Corporation, 1982.

4. Bassett, William G., *A Dynamic Simulation and Feedback Control Scheme for the U. S. Marine Corps' Airborne Remotely Operated Device (AROD)*, Master's Thesis, Naval Postgraduate School, Monterey, California, September 1987.

5. Kuo, Benjamin C., *Automatic Control Systems*, Prentice Hall, 1991.

6. Kwakernaak, Huibert, Sivan, Raphael, *Linear Optimal Control System*, Wiley-Interscience, 1972.

7. Kirk, Donald E., *Optimal Control Theory, An Introduction*, Prentice Hall, 1970.

8. Thaler, George J., *Automatic Control Systems*, West, 1989.

9. Lloyd, Scot D., *An Autopilot Design for the United States Marine Corps' Airborne Remotely Operated Device*, Master's Thesis, Naval Postgraduate School, Monterey, California, September 1987.

10. Franklin, Gene F., Powell, J. David, Workman, Michael L., *Digital Control of Dynamic Systems*, Addison-Wesley, 1990.

11. Friedland, Bernard, *Control System Design an Introduction to State-Space Methods*, McGraw-Hill, 1986.

12. Athans, M., "On the Design of P-I-D Controllers Using Optimal Linear Regulator Theory," *Automatica*, v. 7, pp.643-647, September 1971.

13. Burl, Jeffrey B., "Linear Optimal Estimation and Control Notes," Naval Postgraduate School, Monterey, California, September 1991.

14. White, J.E., Phelan, J.R., "Stability Augmentation and Control Decoupling for the Airborne Remotely Operated Device," *Journal of Guidance, Control, and Dynamics, v.14, pp. 176*-183, January-February 1991.

15. Phone conversations with Mr. J. E. White, Sandia National Laboratories, November-December 1992.

16. Weir, R. J., "Aerodynamic Design Considerations for a Free-Flying Ducted Propeller," *Proceedings of the 1988 Atmospheric Flight Mechanics Conference*, AIAA, Washington, DC, pp.720-731, August, 1988.

17. Ogata, Katsuhiko, *Modern Control Engineering*, Prentice Hall, 1990.

18. Wertz, James R., *Spacecraft Attitude Determination and Control*, D. Reidel Publishing Company, Holland, 1986.

19. Maciejowski, J. M., *Multivariable Feedback Design*, Addison-Wesley, 1989.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22314-6145	2
2.	Library, Code 52 Naval Postgraduate School Monterey, California 93943-5100	2
3.	Chairman, Code EC Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, California 93943-5000	1
4.	Prof. Jeffrey B. Burl, Code EC/Bl Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, California 93940-5000	2
5.	Prof. H. A. Titus, Code EC/Ts Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, California 93940-5000	2
6.	Prof. Richard M. Howard, Code AA/Ho Department of Aeronautics and Astronautics Naval Postgraduate School Monterey, California 93940-5100	2
7.	Director of Training and Education MCCDC Code C46 1019 Elliot Road Quantico, Virginia 22134-5027	1
8.	Captain Joseph P. Davis 102 Greenfield Road Stafford, Virginia 22554	2
9.	Mr. J. E. White, 9132 P.O. Box 5800 Albuquerque, New Mexico 87185-5800	1

Thesis
D17153 Davis
c.1 The design of a robust autopilot for the Archy-tas prototype via linear quadratic synthesis.